# BSD MAGAZINE

FOR NOVICE AND ADVANCED USERS

## HERE COMES THE CLOUD

VOL.5 NO.04
ISSUE 04/2012(33)
1898-9144

# TrueNAS™ Storage Appliance: You are the Cloud

With a rock-solid FreeBSD® base, Zettabyte File System support, and a powerful Web GUI, TrueNAS™ pairs easy-to-manage software with world-class hardware for an unbeatable storage solution.

*Expansion Shelves Available*

TrueNAS™ 2U System

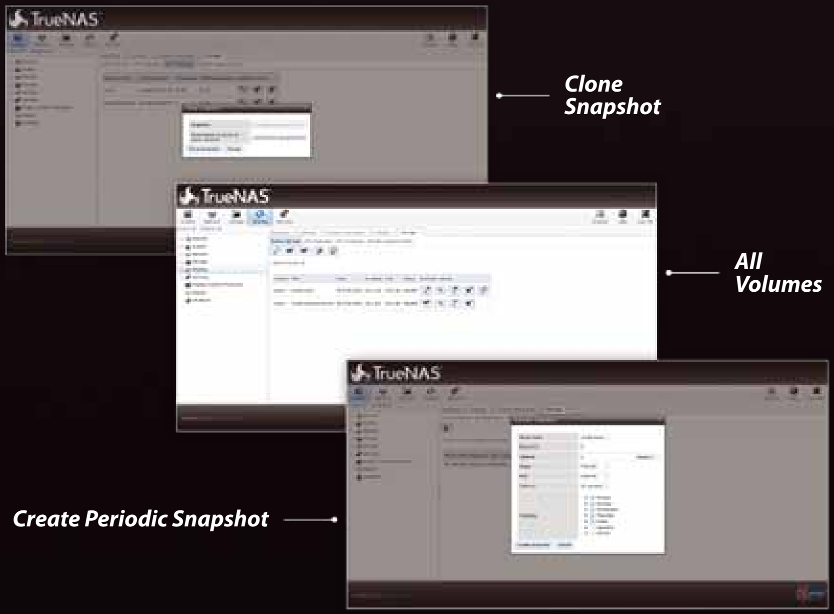
TrueNAS™ 4U System

FreeNAS 8 POWERED

## Storage. Speed. Stability.

In order to achieve maximum performance, the TrueNAS™ 2U and 4U Systems, equipped with the Intel® Xeon® Processor 5600 Series, support Fusion-io's Flash Memory Cards and 10GbE Network Cards. Titan TrueNAS™ 2U and 4U Appliances are an excellent storage solution for video streaming, file hosting, virtualization, and more. Paired with optional JBOD expansion units, the TrueNAS™ Systems offer excellent capacity at an affordable price.

For more information on the **TrueNAS™ 2U** and **TrueNAS™ 4U**, or to request a quote, visit: **http://www.iXsystems.com/TrueNAS**.



## TrueNAS™ 2U KEY FEATURES

- Supports One or Two Quad-Core or Six-Core, Intel® Xeon® Processor 5600 Series
- 12 Hot-Swap Drive Bays - Up to 36TB of Data Storage Capacity*
- Periodic Snapshots Feature Allows You to Restore Data from a Previously Generated Snapshot
- Remote Replication Allows You to Copy a Snapshot to an Offsite Server, for Maximum Data Security
- Software RAID-Z with up to Triple Parity
- 2 x 1GbE Network Interface (Onboard) + Up to 4 Additional 1GbE Ports or Single/Dual Port 10GbE Network Cards

## TrueNAS™ 4U KEY FEATURES

- Supports One or Two Quad-Core or Six-Core, Intel® Xeon® Processor 5600 Series
- 24 or 36 Hot-Swap Drive Bays - Up to 108TB of Data Storage Capacity*
- Periodic Snapshots Feature Allows You to Restore Data from a Previously Generated Snapshot
- Remote Replication Allows You to Copy a Snapshot to an Offsite Server, for Maximum Data Security
- Software RAID-Z with up to Triple Parity
- 2 x 1GbE Network Interface (Onboard) + Up to 4 Additional 1GbE Ports or Single/Dual Port 10GbE Network Cards

**JBOD expansion is available on the 2U and 4U Systems**

** 2.5" drive options available; please consult with your Account Manager*

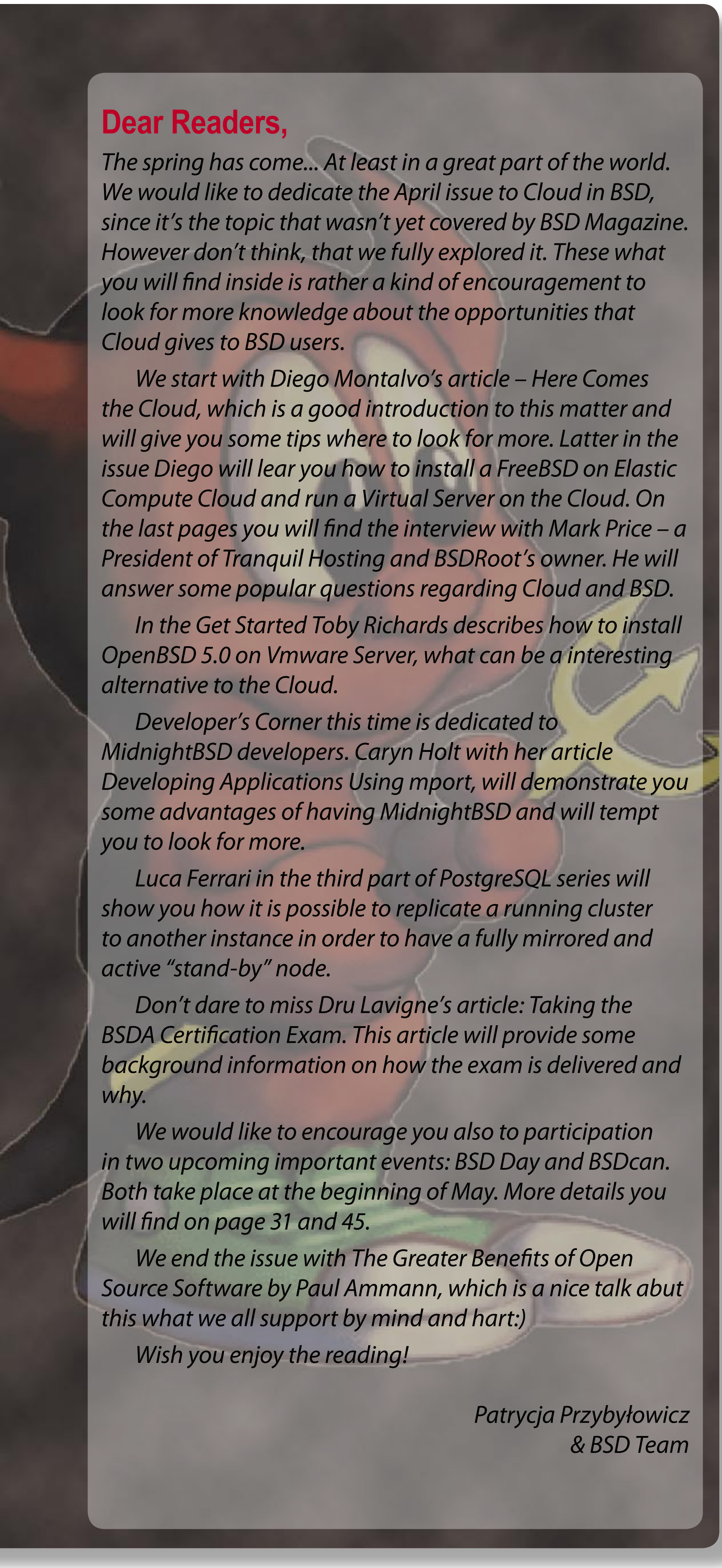
iX systems®

intel® Xeon® inside™

Powerful. Intelligent.

**Call iXsystems toll free or visit our website today!**

**1-855-GREP-4-IX | www.iXsystems.com**



## Dear Readers,

*The spring has come... At least in a great part of the world. We would like to dedicate the April issue to Cloud in BSD, since it's the topic that wasn't yet covered by BSD Magazine. However don't think, that we fully explored it. These what you will find inside is rather a kind of encouragement to look for more knowledge about the opportunities that Cloud gives to BSD users.*

*We start with Diego Montalvo's article – Here Comes the Cloud, which is a good introduction to this matter and will give you some tips where to look for more. Latter in the issue Diego will lear you how to install a FreeBSD on Elastic Compute Cloud and run a Virtual Server on the Cloud. On the last pages you will find the interview with Mark Price – a President of Tranquil Hosting and BSDRoot's owner. He will answer some popular questions regarding Cloud and BSD.*

*In the Get Started Toby Richards describes how to install OpenBSD 5.0 on Vmware Server, what can be a interesting alternative to the Cloud.*

*Developer's Corner this time is dedicated to MidnightBSD developers. Caryn Holt with her article Developing Applications Using mport, will demonstrate you some advantages of having MidnightBSD and will tempt you to look for more.*

*Luca Ferrari in the third part of PostgreSQL series will show you how it is possible to replicate a running cluster to another instance in order to have a fully mirrored and active "stand-by" node.*

*Don't dare to miss Dru Lavigne's article: Taking the BSDA Certification Exam. This article will provide some background information on how the exam is delivered and why.*

*We would like to encourage you also to participation in two upcoming important events: BSD Day and BSDcan. Both take place at the beginning of May. More details you will find on page 31 and 45.*

*We end the issue with The Greater Benefits of Open Source Software by Paul Ammann, which is a nice talk abut this what we all support by mind and hart:)*

*Wish you enjoy the reading!*

*Patrycja Przybyłowicz*
*& BSD Team*

MAGAZINE BSD

**Editor in Chief:**
Patrycja Przybyłowicz
*patrycja.przybylowicz@software.com.pl*

**Contributing:**
Dru Lavigne, Diego Montavo, Toby Richards, Caryn Holt, Luca Ferrari, Paul Ammann

**Top Betatesters & Proofreaders:**
Paul McMath, Bjorn Michelsen, Barry Grumbine, Imad Soltani, Eric De La Cruz, Luca Ferrari, Shayne Cardwell, Norman Golish, Robert Novak, Sander Reiche

**Special Thanks:**
Denise Ebery
Dru Lavigne

**Art Director:**
Ireneusz Pogroszewski

**DTP:**
Ireneusz Pogroszewski

**Senior Consultant/Publisher:**
Paweł Marciniak *pawel@software.com.pl*

**CEO:**
Ewa Dudzic
*ewa.dudzic@software.com.pl*

**Production Director:**
Andrzej Kuca
andrzej.kuca@software.com.pl

**Executive Ad Consultant:**
Ewa Dudzic
*ewa.dudzic@software.com.pl*

**Advertising Sales:**
Patrycja Przybyłowicz
*patrycja.przybylowicz@software.com.pl*

**Publisher :**
**Software Press Sp. z o.o. SK**
**ul. Bokserska 1, 02-682 Warszawa**
**Poland**
worldwide publishing
tel: 1 917 338 36 31
*www.bsdmag.org*

Software Press Sp z o.o. SK is looking for partners from all over the world. If you are interested in cooperation with us, please contact us via e-mail: *editors@bsdmag.org*

# Here Comes the Clouds

In the past couple of years the term "cloud" has been on every tech news headline from companies offering "cloud" computing or start-ups running killer "cloud" based services. After a lot of thinking and rewriting I decided "cloud" can mean different things, it all depends on how it is being referenced. For example "cloud" by itself is short for Internet or long for web.

I think it is important to differentiate between cloud "computing" and "services". "Cloud computing" is a marketing term for virtual hosting services, which allows virtual administration via an Internet connection. "Cloud services" is a fancy phrase for services on the Internet.

In 2006 cloud computing began it's popularity with the debut of Amazon's *Elastic Compute Cloud* (EC2) which began offering virtual servers to a mass audience.

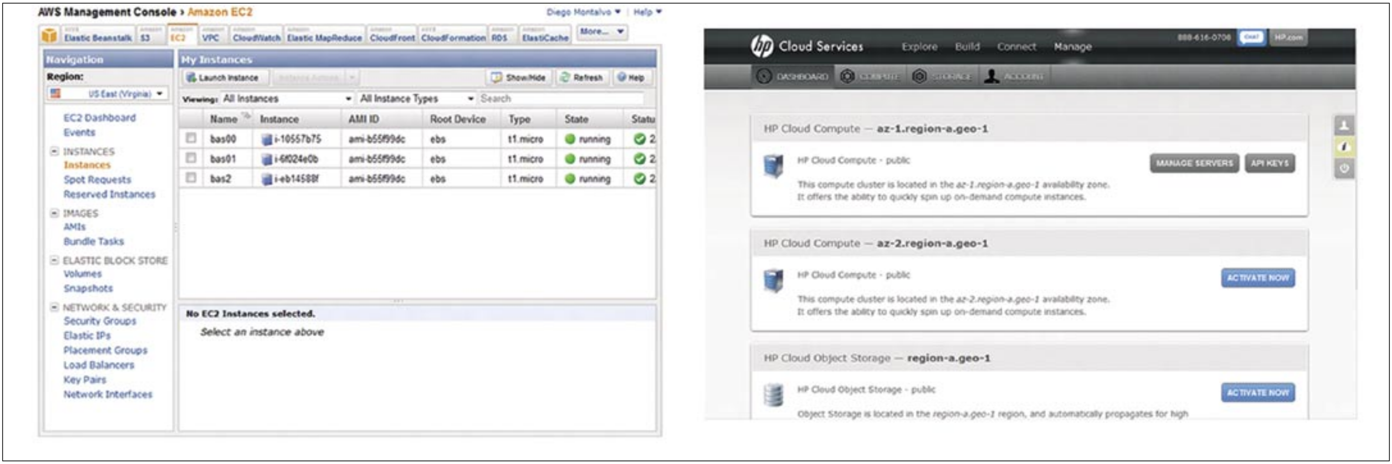
Cloud services have been around since the common non-techie even knew what email was. As early as 1996 Hotmail (*running on FreeBSD*) introduced what initially was a cloud-based email system.

Today cloud computing is offered as "virtual machines" or "instances" which are both scalable and remotely accessed via a SSH, VNC client or a customized Web Interface of sorts (Figure 1).

*"Cloud computing describes a new supplement, consumption, and delivery model for IT services based on Internet protocols, and it typically involves provisioning of dynamically scalable and often virtualized resources." Wikipedia*

Back in 2000 I remember thinking how cool it would be to build a web based word processor such as Word. Even though it was a good idea back then, it is only today that both sophisticated clouds and web technology make it feasible. Once prominent desktop productivity software has been seeing it's demise due to web based counterparts, examples Google Docs replacing the software office suite.

Just as web services have began to replace software, cloud services have began to replace small server farms and on-site server administrators. In today's fast paced world why purchase more servers and pay a server



| | Name | Instance | AMI ID | Root Device | Type | State | Statu |
|---|---|---|---|---|---|---|---|
| ☐ | bas00 | i-10557b75 | ami-b55f99dc | ebs | t1.micro | ● running | ✓ 2 |
| ☐ | bas01 | i-6f024e0b | ami-b55f99dc | ebs | t1.micro | ● running | ✓ 2 |
| ☐ | bas2 | i-eb14588f | ami-b55f99dc | ebs | t1.micro | ● running | ✓ 2 |

No EC2 Instances selected.

*Select an instance above*

HP Cloud Compute — **az-1.region-a.geo-1**

HP Cloud Compute - public

This compute cluster is located in the *az-1.region-a.geo-1* availability zone. It offers the ability to quickly spin up on-demand compute instances.

HP Cloud Compute — **az-2.region-a.geo-1**

HP Cloud Compute - public

This compute cluster is located in the *az-2.region-a.geo-1* availability zone. It offers the ability to quickly spin up on-demand compute instances.

HP Cloud Object Storage — **region-a.geo-1**

HP Cloud Object Storage - public

Object Storage is located in the *region-a.geo-1* region, and automatically propagates for high

**Figure 1.** *Amazon and HP Cloud Computing Web Interfaces*

administrator when you can simply scale your web services with a few clicks and a credit card.

**Cloud computing is to hardware as cloud services is to software. Both are evolutions of technology and both are replacing there counterpart.**

Today's Internet consists of text, images, video, music, on-demand video, real-time communication, instant stock quotes and more. Even though the Internet basics such as HTML are still the same, what has changed is the magnitude of information being exchanged every second of every day. Data storage has increased enormously. Once upon a time a single server had enough capacity to manage a popular website. Today's popular websites run on hundreds if not thousands of servers and are constantly upgrading equipment. Since the inception of the Internet it has always been fantasized that Network computers would replace software based computers, it is happening now with cloud computers and cloud services.

With the advancement of Internet technology and advanced web services, more and more users are becoming more reliant on cloud services than on computer software.

Cloud services can be found in the smallest of devices. Smart mobile devices and most mobile applications rely on some sort of data push or data request from the cloud to add application functionality (Figure 2).

Most mobile carriers currently offer secure cloud services for storing a mobile user's contacts and other stored information. Recently Apple Inc. has been pushing iCloud, a backup service which pushes users device content to all Apple devices.

Cloud-computing has also began to infuse itself into different BSD flavors: Amazon's Elastic Compute Cloud (EC2) offers users the choice of FreeBSD and NetBSD AMIs (Amazon Machine Instances). Where as RootBSD offers users a choice of BSD on Xen based virtual private services with full technical support.

In 2006 Amazon *Web Services* (AWS) released EC2 which opened up Amazon's own server infrastructure as a way of making use of it's unused server capacity. EC2 allows users to choose from hundreds of AMIs including different tier offerings of FreeBSD and NetBSD. EC2 also provides users with an administration web console. AMI customization can be done using any SSH client such as PuTTY or terminal.

RootBSD is a dedicated BSD virtual server provider which specializes in FreeBSD but offers other BSD distributions upon request. RootBSD offers users flat-rate monthly plans. RootBSD virtual servers are Xen based and allow users to administrate services using any SSH

gatesfoundation Via @getschooled: Nominate a student or teacher amazing at math for a chance go to NYC to ring the NYSE bell: http://bit.ly/9PzOmP

Samasource Got serious coding chops? Want to be a force for good? Come hack for @samasource as a web dev: http://su.pr/9nblvL. Please RT!

FCC Questioning Verizon CEO's recent spectrum comments, noting their longstanding support. CoS Edward Lazarus http://bit.ly/burl78

Techmeme iPhone OS 4 keynote video now online (@futurepaul / Engadget) http://techme.me/=OI6

**Figure 2.** *Mobile Applications and Cloud Interaction*

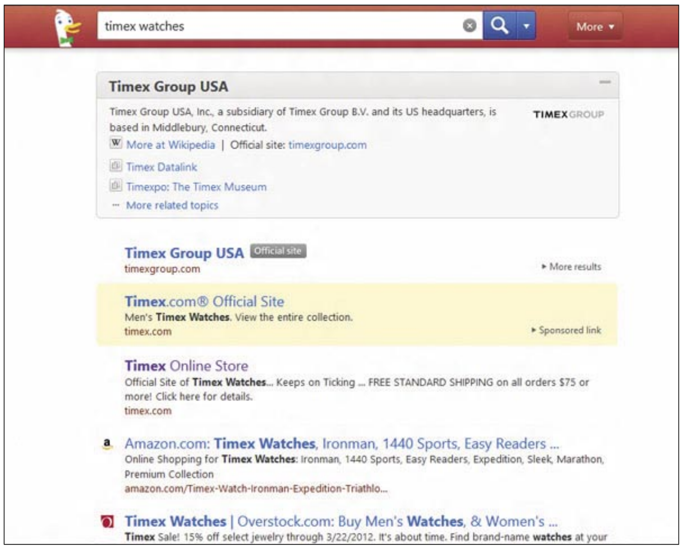
client. RootBSD offers technical support for all VPS or custom plans.

BuildaSearch (BaS) a web service I founded is 100% cloud based, running on FreeBSD Xen based virtual servers. BaS allows users to crawl and index thousands of pages in real-time. Cloud computing allows BaS to scale services with a few clicks.

*DuckDuckGo* (DDG) a search engine which emphasizes privacy, uses a cloud infrastructure to power it's growth. DDG results are a compilation of many sources, including Yahoo! Search BOSS, Bing, Wikipedia, Wolfram Alpha and its own web crawler. DDG uses FreeBSD for crawling and team coordination (Figure 3).

Cloud-computing in the past 6 years has grown from an experimental concept at Amazon to a viable business model which is reshaping information technology as a whole. Today cloud computing powers sites such as Twitter, DuckDuckGo, Digg, Crunchbase, BuildaSearch and even Amazon's own enormous market-place.

In reality the first clouds are already here, beginning to influence our everyday lives from accessing social sites to filing taxes over the web. As Internet technologies evolve, bigger and more sophisticated clouds will appear. Even though the clouds today seem large, they are initially just the beginning to a world full of clouds which in the not to distant future will include a cloud toaster and crock-pot which tweets when the roast is ready.



Timex Group USA

Timex Group USA, Inc., a subsidiary of Timex Group B.V. and its US headquarters, is based in Middlebury, Connecticut.
More at Wikipedia | Official site: timexgroup.com
Timex Datalink
Timexpo: The Timex Museum
More related topics

Timex Group USA Official site
timexgroup.com
More results

Timex.com® Official Site
Men's **Timex Watches**. View the entire collection.
timex.com
Sponsored link

**Timex** Online Store
Official Site of **Timex Watches**... Keeps on Ticking ... FREE STANDARD SHIPPING on all orders $75 or more! Click here for details.
timex.com

Amazon.com: **Timex Watches**, Ironman, 1440 Sports, Easy Readers ...
Online Shopping for **Timex Watches**: Ironman, 1440 Sports, Easy Readers, Expedition, Sleek, Marathon, Premium Collection
amazon.com/Timex-Watch-Ironman-Expedition-Triathlo...

**Timex Watches** | Overstock.com: Buy Men's **Watches**, & Women's ...
**Timex** Sale! 15% off select jewelry through 3/22/2012. It's about time. Find brand-name **watches** at your

**Figure 3.** *DuckDuckGo Search Engine*

## References

- *http://www.wikipedia.org*
- *http://hpcloud.com/*
- *https://www.icloud.com*
- *http://aws.amazon.com*
- *http://rootbsd.net*
- *http://buildasearch.com*
- *http://duckduckgo.com*
- *http://xen.org/*

## DIEGO MONTALVO

***Diego Montalvo is the founder of BuildaSearch.com a (site search engine web service) as well as Urloid.com a (URL shortening service). Diego by trade is a web developer and technical writer who enjoys the beach, socializing, staying up late, exercising and an drinking an occasional pint. Feel free to contact Diego at diego@earthoid.com.***

database pdf

XML

<basresult><title>An SQL White Paper June 2011 CoreDB Inc. </title>
<longurl>http://www.coredbinc.com/us/products/database/039450.pdf</longurl>
<basummary>An CoreDB White Paper June 2011 DB <b>Database</b>:
The <b>Database</b> of Choice for  Deploying Scalable Solutions CoreDB<b>
Database</b>: the <b>Database</b> of Choice for ......</basummary>
<showurl>http://www.coredbinc.com/us/products/database/039450.pdf
</showurl><size>881904</size>
<modified>Mon, 19 Mar 2012, 02:08:34 UTC</modified>
<type>application/pdf</type></bsresult>

JSON

{"title": "An SQL White Paper June 2011 CoreDB Inc. ",
"longurl": "http://www.coredbinc.com/us/products/database/039450.pdf",
"basummary": "An CoreDB White Paper June 2011 DB <b>Database</b>:
The <b>Database</b> of Choice for  Deploying Scalable Solutions CoreDB<b>
Database</b>: the <b>Database</b> of Choice for ......",
"showurl": "http://www.coredbinc.com/us/products/database/039450.pdf",
"size": "881904", "modified": "Mon, 19 Mar 2012, 02:08:34 UTC",
"type": "application/pdf"}

pdf html doc

scalable site search engine

hosted search technology

web control panel

open format search API

{free}

buildasearch
www.buildasearch.com

# Developing

## Applications Using mport

In the February issue, you saw how to use the mport system as an end user. In MidnightBSD, you can access the mport features as a developer and add support for mport to your existing C, C++ or Objective C application.

**What you will learn…**

- Add basic mport management to existing C, C++ or Objective C applications

**What you should know…**

- C, C++ or Objective C

When you start to work with mport, any function that is marked as MPORT_PUBLIC_API is considered safe for external applications to use. Feel free to use any function however the public API functions are the only ones guaranteed to work. To see if a function is marked as public, please look at it's implementation.

The include file is *mport.h*.

Initially, the mport API was developed to allow MidnightBSD developers to rapidly create new mport tools.

However as discussions about new port tools continued that eventually other developers would want to add mport functionality to their applications.

**Listing 1.** *Creating and initializing mport instance*

```
mportInstance *mport;
mport = mport_instance_new();

if (mport_instance_init(mport, NULL) != MPORT_OK) {
    ...
}
```

**Listing 2.** *Looking up package name*

```
const char *packageName = "test pkg";
mportIndexEntry **indexEntries;

if (mport_index_lookup_pkgname(mport, packageName,                          &indexEntries) != MPORT_OK) {
...
}
```

At this time, MidnightBSD only has a C mport API however we would like to add better scripting support. In particular, we want to add Python support.

## Initializing mport

For any mport call, you need to create a new mport_ instance. In the sample code, I created a mportInstance

**Listing 3.** *Reading and display mport information*

```
int info(mportInstance *mport, const char *packageName) {
   mportIndexEntry **indexEntry;
   mportPackageMeta **packs;
   char *status, *origin;

   if (packageName == NULL) {
      fprintf(stderr, "Specify package name\n");
      return 1;
   }

   indexEntry = lookupIndex(mport, packageName);
   if (indexEntry == NULL || *indexEntry == NULL) {
      fprintf(stderr, "%s not found in index.\n", packageName);
      return 1;
   }

   if (mport_pkgmeta_search_master(mport, &packs, "pkg=%Q", packageName) != MPORT_OK) {
      warnx("%s", mport_err_string());
      return 1;
   }

   if (packs == NULL) {
      status = "N/A";
      origin = "";
   } else {
      status = (*packs)->version;
      origin = (*packs)->origin;
   }

   printf("%s\nlatest: %s\ninstalled: %s\nlicense: %s\norigin: %s\n\n%s\n",

       (*indexEntry)->version,
       status,
       (*indexEntry)->license,
       origin,
       (*indexEntry)->comment);

   mport_index_entry_free_vec(indexEntry);
   return 0;
}
```

pointer and creating the instance by calling `mport_instance_new()`.

After I created the mport instance, I initialized it using `mport_instance_init()`. If the call was successful, it will return MPORT_OK.

## Working with the mport index

For several tasks, you will first need to load the mport index. To load the index call `mport_index_load()`.

```
int result = mport_index_load(mport);
```

A successful index load will return MPORT_OK.

To install, update or delete a package, you will need to lookup the package in the index.

In later code examples, you will see how to use the index entries. The structure for mportIndexEntry is defined in *mport.h*. To free an index entry, use `mport_index_entry_free_vec()`.

## Installing and deleting packages

You will need to have loaded the mport index to install a port.

```
const char *packageName
char *buf, *packagePath;
mportIndexEntry **indexEntry;
```

After looking up the package in the index, construct the package path. `MPORT_LOCAL_PKG_PATH` is the local downloads location. For default installations, it should be */var/db/mport/downloads*.

```
asprintf(&packagePath, „%s/%s", MPORT_LOCAL_PKG_PATH,
 (*indexEntry)->bundlefile);
```

To check if the package exists locally, call `mport_package_exists()`.

If you need to download the package, call `mport_fetch_bundle()`. If the download was successful, MPORT_OK is returned.

```
mport_fetch_bundle(mport, (*indexEntry)->bundlefile)
```

Additionally, you can verify the package by calling `mport_verify_hash()`. The hash for the package is in the `mportIndexEntry`.

To install the package, call `mport_install_primative()`. The function `mport_delete_primative()` removes a package.

## On the Web

- *http://www.midnightbsd.org* – MidnightBSD project's website
- *http://www.midnightbsd.org/documentation/mports/* – mport documentation

## Glossary

- mport

## Updating an existing package

You can update the package and optionally any dependencies. For this example, we'll only be updating the package.

Similar to installing a package, you would lookup the index entry and construct the package path. Instead of calling `mport_install_primative()`, you would call `mport_update_primative()`.

## Other useful mport functions

Another important mport structure is mportPackageMeta. This structure contains additional information such as langauge and package categories. Also to get up or down dependencies (`mport_pkgmeta_get_updepends()` and `mport_pkgmeta_get_downdepends()`) for a given package, you will need to have the meta information.

The mport library includes two error reporting functions – `mport_err_code()` which returns the error code as an integer and `mport_err_string()` which returns the error message.

## Summary

The mport library allows developers to integrate some or all of mport features into their own applications without calling `exec()`. This article is just an introduction to the many features available to MidnightBSD application developers. I would recommend going to the MidnightBSD website for further information.

**CARYN HOLT**

***Caryn Holt is a MidnightBSD developer and software engineer for Rovi in Ann Arbor, Michigan.***

MidnightBSD

*the BSD for everyone*

**Come join MidnightBSD:**

• desktop BSD •
• over 2,000 ports •
• unique mports package installer •
• helpful community •
• hands-on learning for new developers •

**www.midnightbsd.org**

# Taking the BSDA Certification Exam

The first article in this series (in the February 2012 issue) addressed some common misconceptions about certification and described why you should be BSDA certified. The second article in this series (in the March 2012 issue) discussed how to prepare for the BSDA certification exam. This article will provide some background information on how the exam is delivered and why. It will then describe where to take the exam and how to arrange for an exam if there currently isn't an examination event or testing center near your location.

The BSDA certification exam became available in February, 2008. The *BSD Certification Group* (BSDCG) had several goals in mind when launching this examination:

- Maintain the psychometric validity of the exam.
- If possible, use BSD operating systems and open source software for exam delivery.
- Keep the exam price as globally affordable as possible.
- Make the exam available to anyone, regardless of their location.

Since these goals impact on how the exam is delivered, let's take a closer look at each:

## Maintain the Psychometric Validity of the Exam

Assessing practical, real-world system administration skills is an integral component of the BSDA examination. A lot of work goes into the exam creation process to ensure that the resulting certification is psychometrically valid. To maintain the validity of the examination, certain requirements need to be followed when the exam is taken. For example:

- the identity of the person taking the exam must be verified using government issued, photo identification. This is to ensure that the person taking the exam is who they say they are and to prevent someone from taking the exam for someone else.
- the person taking the exam needs to be monitored during the exam to ensure that they don't have access to additional information sources, tamper with the exam materials, copy exam questions, or remove exam materials from the exam room.

An assessment is not accurate if the person is not who they claim to be or if a person who doesn't have the skills needed to pass the exam finds a way to cheat the assessment. This is the reason why exam candidates need to have their ID checked and why their activity must be monitored when taking the exam. The person doing the checking and monitoring is the proctor and they must be trusted by the organization which provides the exam.

The requirement to use a proctor places restrictions on how the exam can be delivered. For example, we often hear the question "why can't I take the exam online from home?". This type of exam delivery is hard to proctor for several reasons. Verifying the person's ID requires either a photocopy or viewing the ID on a webcam, making it difficult to obtain a clear image or to spot a counterfeit. Monitoring is not as reliable: a webcam can be pointed at the exam taker, but it won't notice if the person is referring to notes outside the camera view, has found a way to subvert the exam application and access additional resources such as an Internet search in a browser, or is running screen capture software to copy the contents of the exam. It also requires the proctor to have access to the camera view or to the user's desktop session – this isn't as scalable as having a proctor monitor a room full of examinees in person.

## Use Open Source Software, Keep the Exam Price Globally Affordable, and Make the Exam Available to Anyone

When it comes to exam delivery, these three goals are related. Commercial solutions which provide proctored testing centers throughout the world exist (the best known examples are VUE and Prometric), but they can be problematic for several reasons:

- the testing software does not use open source software (it is usually Microsoft Windows, flash, and Internet Explorer based). This limits the types of exam questions and makes providing interactive lab scenarios difficult. While it is possible to create interactive flash scenarios, these are expensive to develop and don't provide the same flexibility as an operating system running in a virtual environment or a FreeBSD jail.
- these solutions assume high delivery volume and charge accordingly, making it difficult to provide an affordable exam. For example, there is an annual fee (typically in the high, five-figure US dollar range) that must be paid every year, regardless of the number exams delivered. If the testing organization doesn't deliver enough exams to cover their annual fee, a financial loss is incurred for that year.
- a psychometrically valid exam undergoes constant statistical analysis to determine if any questions need to be modified (e.g. they are determined to be too easy or too hard). Commercial solutions charge a publication fee (typically in the low, four figure US dollar range) whenever exam questions are changed or whenever new exams become available, providing a financial dis-incentive for keeping the exam up-to-date or providing additional versions of an exam.
- the larger testing companies tend to have testing centers located in most countries throughout the world, but charge the largest annual fee. Smaller companies charge a lower annual fee, but tend to have good North American coverage and limited locations in other parts of the world.

When determining if a commercial test delivery solution is a good match for an organization that provides an exam, one needs to balance the number of people expected to take the exam in a year, where those people are located, and how much they can afford to pay to take the exam. If the number of examinees is low (less than several thousand per year), examinees tend to live in areas that aren't near a testing center, or examinees tend to be in countries where an exam fee in the $200 USD range (the average price for an exam) is unaffordable, alternative ways to deliver the exam need to be explored.

Before the launch of the BSDA exam, the BSDCG launched a Test Delivery Survey to help determine the testing needs of the BSD system administrators community. The report on that survey is available at *http://www.bsdcertification.org/downloads/delivery_survey.pdf*. One of the conclusions of the report is that "70% of testing candidates are unwilling to pay more than $100 USD to take the exam; the cost of using a test delivery solution will have to fall below this price point and still allow for the costs of psychometric analysis and administration of the BSDCG organization". The survey results also indicated that "testing candidates are scattered throughout the globe" (58 countries were represented in the survey) and that "the majority of testing candidates are willing to travel to take the exam".

When the BSDA was launched in 2008, there weren't any existing open source test delivery solutions and the annual fee imposed by the existing commercial solutions was beyond the starting budget of the BSDCG. The decision was made to offer a paper based version of the exam at hosted events around the world and to further research the feasability of either a home-grown open source solution or a more reasonably priced commercial solution.

At this time, there are now two delivery methods for taking the exam: a paper based exam at an exam event or a computer based exam at a testing center. The content of the BSDA is the same, regardless of the delivery method.

## Paper Based Exams

The first exam event was held during the Southern California Linux Expo in 2008. Since then, over 120 exam events have been organized at technical conferences, schools, and places of employment throughout the world. You can view the upcoming and past events at *https://register.bsdcertification.org//register/events*.

Taking the paper-based exam at an exam event offers several advantages:

- it provides the opportunity to meet and network with other system administrators who are also interested in BSD
- since the costs to deliver the exam are primarily the cost of shipping the exams to and from the event, the price of the exam can be kept at the mostly global affordable price of $75 USD

There are also some limitations to this method of exam delivery:

- there are only so many events and locations per year.
- organizing and advertising upcoming exam events relies on the assistance of the community.

Efforts by a local community to arrange and promote exam events in their geographic area directly impact on the success of an exam event. In turn, successful exam events benefit the local system administrator community and help to promote the use of BSD in that geographic area.

## Arranging for an Exam Event

There are several advantages to arranging an exam event in your city:

- you don't have to wait until an exam event comes to a location near you.
- as a member of your geographic community, you have a bettter idea of which local resources are available for hosting an exam event.
- it provides a networking opportunity to find and associate with other BSD system administrators.
- the time leading up to the event provides a study opportunity to meet in person and help each other learn the exam objectives.

If you are interested in seeing an exam event organized in your city, check to see if your employer, a loca l educational institution or training center, your local user group, or an upcoming technical conference is interested in hosting an exam event.

In order to host an event, the interested organization needs to be able to provide:

- a quiet room that can comfortably sit 6-8 people, not too close together. Internet is not needed as the exam is paper based. A suitable room is typically easy to arrange with an employer or school. If the event is being organized by a user group that doesn't have their own facility, eheck with the local library or city hall to see if it is possible to reserve a room in a municipal building. If you are contacting a conference organizer, ask if you can reserve one of the conference rooms for a period of 2 hours either during the lunch hour or at the end of the conference day.
- a trusted person to act as the proctor. The person to act as proctor should either be: known in the BSD community, a teacher at an accredited educational institution, a trusted employee, or a speaker at a conference. The proctor will be required to adhere to an NDA to protect the integrity of the exam and can not be certified in the exam that they are proctoring. Depending upon the location, the BSDCG may already know of a proctor who lives close by or who is able to travel to the event. The BSDCG can also assist you in finding a suitable proctor.
- 6-8 weeks notice to give the BSDCG time to advertise the event and ship the exams.
- aim for at least 4 people interested in taking the exam at the event. Talk to your coworkers, fellow students, user group members, or use social media to see if you can drum up some interest.

Once you find an organization willing to host an exam event, contact *chair@bsdcertification.org* with the details about the location and date. The BSDCG will work with you to make sure a suitable proctor is available, that the event is added to the registration website and advertised through social media, and that the exams are shipped to arrive in time for the event.

## Computer Based Exams

Beginning in April, 2011, the BSDCG partnered with SMT to offer a computer based version of the BSDA at IQT testing centers (you can read the press release with the details at *http://bsdcertification.org/news/pr061.html*). This partnership provides several advantages:

- you don't have to wait for an exam event as you can schedule your exam at any time.
- you don't have to wait to receive your exam results as your score report is printed for you at the end of the exam.

Depending upon where you live, there are some limitations to this exam delivery method:

- most testing centers are in North America, though there are centers located outside of North America. The list of testing center locations is here: *http://www.isoqualitytesting.com/mlocations.aspx*.
- the exam price is higher in order to cover the costs of using the testing center network. The price to take the exam at a testing center is $150 USD, which is still affordable in North America and Western Europe but which may not be affordable in some parts of the world.

If you run a testing center or know of a testing center in your city who would like to be added to the IQT testing network, contact *chair@bsdcertification.org* with the details. We are especially interested in adding at least one testing center in Russia as this country has many people interested in BSD certification.

## Registering for an Exam

In order to take the BSDA exam you must first register for a BSDCG ID at *https://register.bsdcertification.org//register/get-a-bsdcg-id*. Once you have an ID, check the events page (*https://register.bsdcertification.org//register/register-for-an-exam*) to see if there is an exam event or testing center location near you. If there isn't, let us know if an organization in your area is interested in hosting an exam so that we can add it to the events page.

If you select a paper exam event, the proctor will be notified of your registration so that they knows who to expect on exam day and so that they can notify you if there is a room change.

If you select a computer based exam, you will not be able to schedule your exam until after payment is received. After making your payment, a link will be emailed to you with the information that you will need to schedule an exam. You have up to one year to schedule the exam after making your payment.

Most exam payments are made through PayPal. If you need an invoice for your payment or are unable to pay using PayPal, send an email to *register@bsdcertification.org*.

## Summary

This article described the exam delivery methods for the BSDA exam, how to arrange for an exam event, and how to register for an exam.

In the June issue of BSD Mag this series will continue by describing how exam questions are created and how interested system administrators can contribute to the exam creation and review process.

**DRU LAVIGNE**

***Dru Lavigne is author of BSD Hacks, The Best of FreeBSD Basics, and The Definitive Guide to PC-BSD. As Director of Community Development for the PC-BSD Project, she leads the documentation team, assists new users, helps to find and fix bugs, and reaches out to the community to discover their needs. She is the former Managing Editor of the Open Source Business Resource, a free monthly publication covering open source and the commercialization of open source assets. She is founder and current Chair of the BSD Certification Group Inc., a non-profit organization with a mission to create the standard for certifying BSD system administrators, and serves on the Board of the FreeBSD Foundation.***

# Installing OpenBSD 5.0 on VMware Server

We're going to install OpenBSD 5.0. With the information in this article, you'll learn how to install it both on your own computer and via VMware Server. For my example, I'll use my own Bsdvm.com account.

**What you will learn…**

- How to install a new instance of OpenBSD 5.0.

**What you need: One of**

- A computer and OpenBSD 5.0 CD or
- A Bsdvm.com account and a Windows computer (a limitation of VMware Server)
- Your own VMware Server 2.0 instance.

As far as I know, Bsdvm.com is the only hosting provider that provides console access to your *virtual private server* (VPS). If you intend to install OpenBSD 5.0 on your own computer, then just skip past step 5.

## Step 1

Get an account with *Bsdvm.com*.

## Step 2

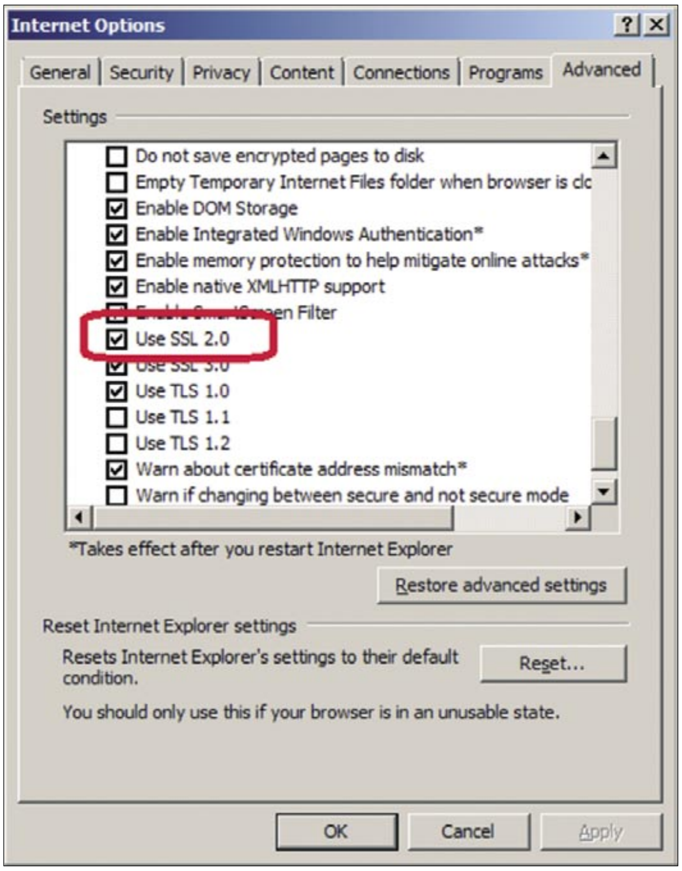
E-mail *support@bsdvm.com*, and ask them for two things: to insert the BSD 5.0 ISO into your Virtual Machine and to give you information for accessing the console.

## Step 3

Configure Internet Explorer to accept SSL 2.0. These are limitations of VMware Server. The VMware plugin (a Windows .exe file) used to work with Firefox, but not since they started accelerating the incrementation of version numbers. VMware has also not upgraded their Server product to use anything better than SSL 2.0.

## Step 4

Browse to the address given by Bsdvm.com support staff, and log on. Figure 2 shows how to find the console screen, which will prompt you to install the console plugin.



Settings

- ☐ Do not save encrypted pages to disk
- ☐ Empty Temporary Internet Files folder when browser is clc
- ☑ Enable DOM Storage
- ☑ Enable Integrated Windows Authentication*
- ☑ Enable memory protection to help mitigate online attacks*
- ☑ Enable native XMLHTTP support
- ☑ Enable SmartScreen Filter
- ☑ Use SSL 2.0
- ☑ Use SSL 3.0
- ☑ Use TLS 1.0
- ☐ Use TLS 1.1
- ☐ Use TLS 1.2
- ☑ Warn about certificate address mismatch*
- ☐ Warn if changing between secure and not secure mode

*Takes effect after you restart Internet Explorer

Reset Internet Explorer settings

Resets Internet Explorer's settings to their default condition.

You should only use this if your browser is in an unusable state.

**Figure 1.** *Internet Explorer Settings*

## Step 5

Notice the power off/pause/power on/reset buttons that have the universal icons, and are available to you. Once you have your console session going, use F2 during the POST to get into BIOS. Set the VM to boot to CDROM first.

Now we're ready to start installing OpenBSD 5.0. Whenever there is a default answer, pressing ENTER without any input will accept that default answer (which you'll see in brackets) One by one, here are the questions that you'll be asked:

```
(I)nstall, (U)pgrade, or (S)hell?
Answer "i" to install OpenBSD 5.0
```

Choose your keyboard layout.
Most people can accept the default layout. Press "?" for more options.

System hostname?
The answer is up to you.

Which network interface do you want to configure? [vic0]
Press ENTER to accept the default of `vic0`.

You will get the answers to the following questions from *support@bsdvm.com*:

- IPv4 address
- Netmask
- IPv6 address (probably none)

Now you'll be asked which other network interface you want to configure. Press *ENTER* to accept the default of none.

Default IPv4 route?
Get the answer from *support@bsdvm.com*.

DNS Domain Name?
That's up to you.

DNS Nameservers?
Get the answer from *support@bsdvm.com*.

Do you want to do any manual network configuration? [no]
No.

Password for root account?
That's up to you. You'll have to confirm it.

Start sshd(8) by default? [yes]
Yes. That's how we're going to access the system after the install is done.

Start ntpd(8) by default? [no]
Yes. This will keep your time synchronized.

Do you expect to run the X Window System? [yes]
No. Since this tutorial assumes that you have limited resources, we're not going to use X.

Change default console to com0? [no]
No.

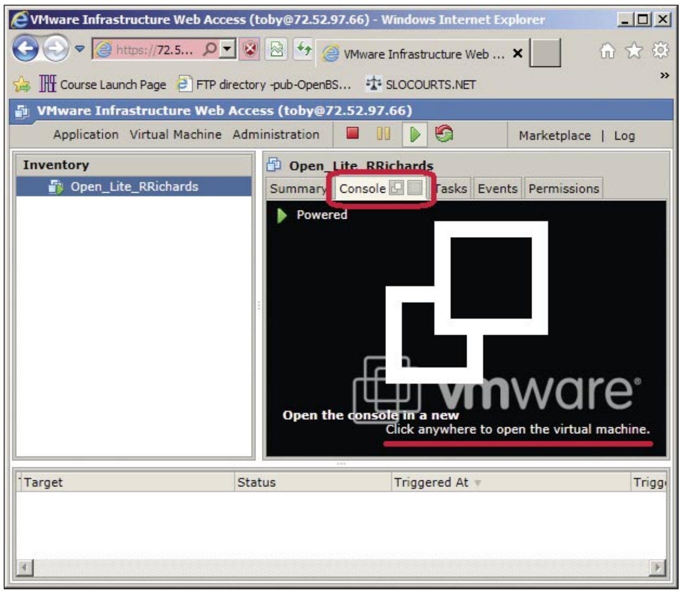
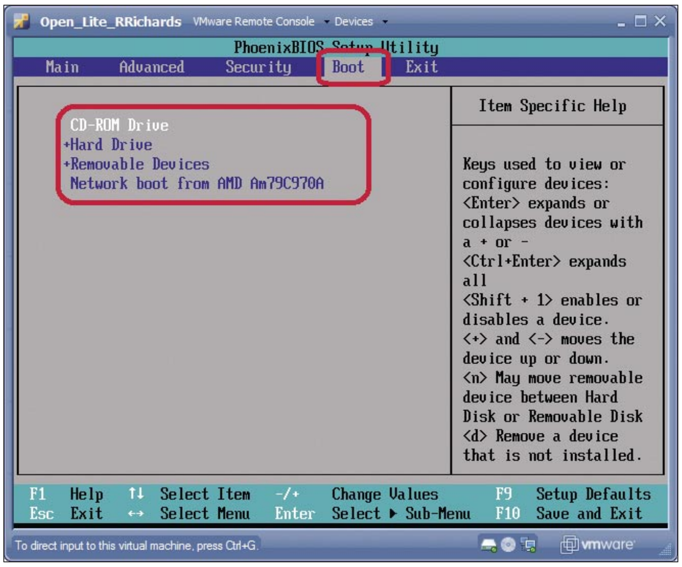

VMware Infrastructure Web Access (toby@72.52.97.66) - Windows Internet Explorer
https://72.5...
VMware Infrastructure Web ...
Course Launch Page
FTP directory -pub-OpenBS...
SLOCOURTS.NET
VMware Infrastructure Web Access (toby@72.52.97.66)
Application
Virtual Machine
Administration
Marketplace
Log
Inventory
Open_Lite_RRichards
Open_Lite_RRichards
Summary
Console
Tasks
Events
Permissions
Powered
vmware
Open the console in a new
Click anywhere to open the virtual machine.
Target
Status
Triggered At
Trigg

**Figure 2.** *The Console Access Page*



```
                    PhoenixBIOS Setup Utility
  Main    Advanced    Security    Boot    Exit

  CD-ROM Drive                                 Item Specific Help
 +Hard Drive
 +Removable Devices                        Keys used to view or
  Network boot from AMD Am79C970A          configure devices:
                                           <Enter> expands or
                                           collapses devices with
                                           a + or -
                                           <Ctrl+Enter> expands
                                           all
                                           <Shift + 1> enables or
                                           disables a device.
                                           <+> and <-> moves the
                                           device up or down.
                                           <n> May move removable
                                           device between Hard
                                           Disk or Removable Disk
                                           <d> Remove a device
                                           that is not installed.

 F1   Help   ↑↓  Select Item   -/+    Change Values         F9   Setup Defaults
 Esc  Exit   ↔   Select Menu   Enter  Select ▶ Sub-Menu     F10  Save and Exit
```

**Figure 3.** *VMware Client BIOS*

Setup a user (enter a lower-case login name or ‘no’) [no]
Type in a user name here. Mine is “toby”. Later on, we’ll secure the server so that the root user cannot login over SSH.

Full username for user [toby]
Your full name. I put “R. Toby Richards” here.

Password for toby account?
Type a password. You’ll have to confirm it.

Since you set up a user, disable sshd(8) logins for root? [yes]
Yes. This is definitely a best practice for security.

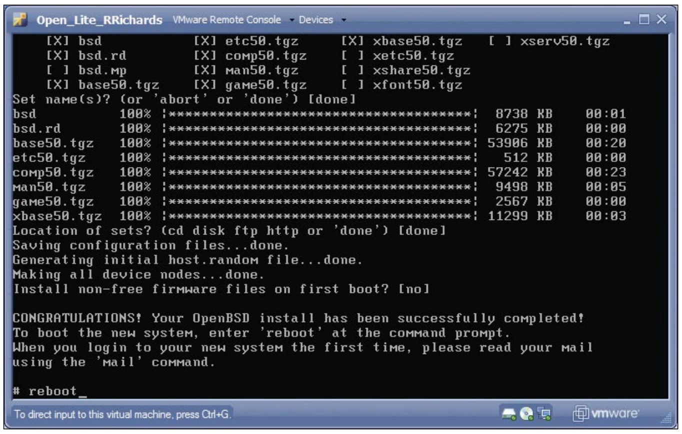
What timezone are you in?
Use the “?” to find your time zone.

Available disks are: sd0 Which one is the root disk (or ‘done’)? [sd0]
sd0

Now we get into disk setup. I’m not going to write out all the text you’ll see. First press “w” to use the whole disk. Then you can press “a” to accept the default partition layout, but if you press “c” for a custom layout, then the procedure for using a single partition (plus a swap partition) is below. We’ll use the following commands:

**Listing 1.** *Single Partition Setup*

```
> z
> a
partition: [a]
offset: [64]
size: [20964761]
FS Type: [4.2BSD]
mount point: [none] /
> m
partition to modify: [] a
offset: [64]
size: [20964736] -512M
FS type: [4.2BSD]
mount point: [/]
> a
partition: [b]
offset: [19920576]
size: [1044249]
FS type: [swap]
> w
> q
```



```
    [X] bsd           [X] etc50.tgz    [X] xbase50.tgz   [ ] xserv50.tgz
    [X] bsd.rd        [X] comp50.tgz   [ ] xetc50.tgz
    [ ] bsd.mp        [X] man50.tgz    [ ] xshare50.tgz
    [X] base50.tgz    [X] game50.tgz   [ ] xfont50.tgz
Set name(s)? (or 'abort' or 'done') [done]
bsd          100% |*************************************|  8738 KB    00:01
bsd.rd       100% |*************************************|  6275 KB    00:00
base50.tgz   100% |*************************************| 53906 KB    00:20
etc50.tgz    100% |*************************************|   512 KB    00:00
comp50.tgz   100% |*************************************| 57242 KB    00:23
man50.tgz    100% |*************************************|  9498 KB    00:05
game50.tgz   100% |*************************************|  2567 KB    00:00
xbase50.tgz  100% |*************************************| 11299 KB    00:03
Location of sets? (cd disk ftp http or 'done') [done]
Saving configuration files...done.
Generating initial host.random file...done.
Making all device nodes...done.
Install non-free firmware files on first boot? [no]

CONGRATULATIONS! Your OpenBSD install has been successfully completed!
To boot the new system, enter 'reboot' at the command prompt.
When you login to your new system the first time, please read your mail
using the 'mail' command.

# reboot_
```

**Figure 4.** *Final Steps of OpenBSD 5.0 Installation*

- `z`: remove all partitions
- `a`: add a partition
- `m`: modify a partition
- `w`: write partition table
- `q`: quit

My numbers are for a 10 GB drive and 512 MB RAM. I don’t want to deal with block numbers, so I first create a partition that uses the whole disk, then I resize that partition to make room for swap. You should have the same amount of swap space as RAM. You’ll see me use “-512M”. You should adjust that number to suit your needs.

Now I’ve got a 9.5 GB root partition, and a 512 MB swap space. Next, we get asked which packages to install. Like I said, we’re not installing X, so when the list is presented, I type “-x*” to unselect the X packages. We do need the xbase package to use ports though, so I then type “+xbase*” You’re pretty well done with setup. You’ll see the system installing your selected packages, and then you can reboot. See Figure 4.

Remember: when you reboot, you’ll need to go back into BIOS with F2 to set the hard drive as the first boot device.

At this point, you’re done. Enjoy your shiny new OpenBSD server.

**TOBY RICHARDS**

***Toby Richards has been a network administrator since 1997. Each article comes straight from the notes that he takes when doing a new project with *BSD. Toby recommends bsdvm.com for your hosting needs because they provide console access to your virtual machine.***

Starting from
8.95$

FreeBSD
OpenBSD
NetBSD

VPS HOSTING

- ✓ UNMETERED bandwidth
- ✓ VNC console
- ✓ instant reimaging
- ✓ native IPv6 network
- ✓ always latest BSDs
- ✓ competent support

http://bsdvm.com

# Installing FreeBSD

## on Amazon AWS EC2 Cloud Services

I have had an AWS account since Amazon first introduced the Elastic Compute Cloud (EC2). But to be honest back in the day it was fairly cryptic to get an instance running, the AWS web interface was in it's infancy and documentation was limited.

**What you will learn…**

- Installing FreeBSD on Elastic Compute Cloud (EC2
- Running a Virtual Server on the Cloud
- Setting up a virtual server using PC-BSD or Windows

**What you should know…**

- Basic knowledge of Terminal or PuTTY
- chmod shell commands
- Basic knowledge of ssh

Even though I did get some sort of Linux instance running, I believe the pricing was abit steep for a lone ranger such as myself.

Since the inception of EC2, folks wanting to run FreeBSD could not, due to some conflicts between the AWS XEN and FreeBSD. On December 2010, however, it was announced that FreeBSD was available on EC2. Moving forward to 2012, I founded BuildaSearch.com a site search service which you guessed it, is powered by FreeBSD. Anyways knowing that FreeBSD could now run on EC2, I decided to build a customized EC2 instance so that I can test out some new BuildaSearch features.

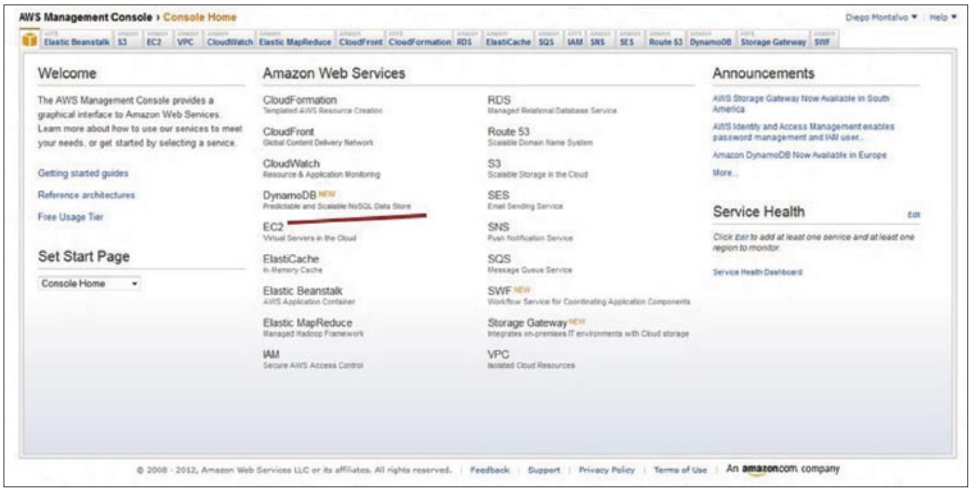
This tutorial will provide the steps needed to have a FreeBSD virtual server on the EC2 using PC-BSD or Windows.

**What you will need: Amazon AWS account, Web-browser, Terminal (PC-BSD), PuTTY, PuTTYgen (Windows)**

## Getting Started

**1.)** (*Windows*) Before we begin with AWS services you will need to download "PuTTy" and "PuTTYgen" from the PuTTY website.

**2.)** Log into your AWS account and click on the *EC2 Virtual Servers in the Cloud* link. (Figure 1)


AWS Management Console › Console Home
Diego Montalvo ▾ | Help ▾
Elastic Beanstalk | S3 | EC2 | VPC | CloudWatch | Elastic MapReduce | CloudFront | CloudFormation | RDS | ElastiCache | SQS | IAM | SNS | SES | Route 53 | DynamoDB | Storage Gateway | SWF

Welcome

The AWS Management Console provides a graphical interface to Amazon Web Services. Learn more about how to use our services to meet your needs, or get started by selecting a service.

Getting started guides
Reference architectures
Free Usage Tier

Set Start Page

Console Home

Amazon Web Services

CloudFormation – Templated AWS Resource Creation
CloudFront – Global Content Delivery Network
CloudWatch – Resource & Application Monitoring
DynamoDB NEW – Predictable and Scalable NoSQL Data Store
EC2 – Virtual Servers in the Cloud
ElastiCache – In-Memory Cache
Elastic Beanstalk – AWS Application Container
Elastic MapReduce – Managed Hadoop Framework
IAM – Secure AWS Access Control
RDS – Managed Relational Database Service
Route 53 – Scalable Domain Name System
S3 – Scalable Storage in the Cloud
SES – Email Sending Service
SNS – Push Notification Service
SQS – Message Queue Service
SWF NEW – Workflow Service for Coordinating Application Components
Storage Gateway NEW – Integrates on-premises IT environments with Cloud storage
VPC – Isolated Cloud Resources

Announcements

AWS Storage Gateway Now Available in South America
AWS Identity and Access Management enables password management and IAM user...
Amazon DynamoDB Now Available in Europe
More...

Service Health — Edit

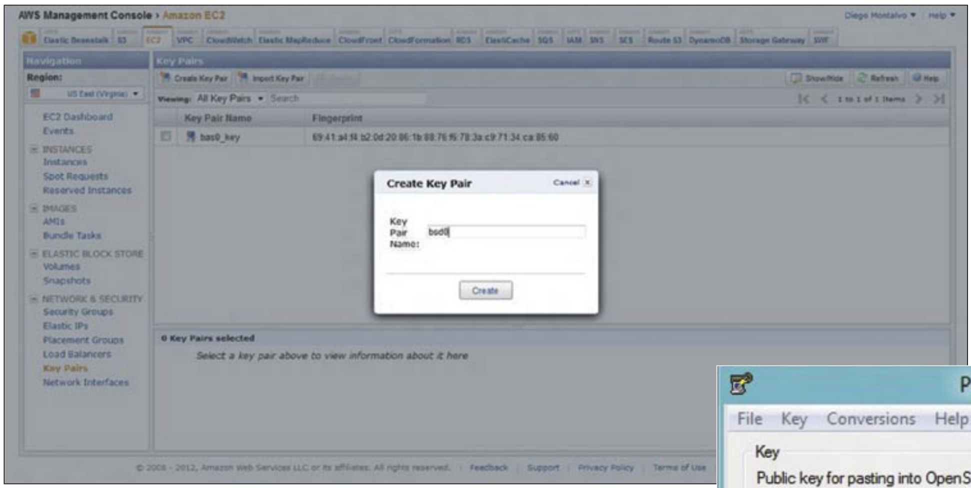
Click Edit to add at least one service and at least one region to monitor.

Service Health Dashboard

© 2008 - 2012, Amazon Web Services LLC or its affiliates. All rights reserved. | Feedback | Support | Privacy Policy | Terms of Use | An amazon.com company

**Figure 1.** *Amazon AWS Management Console*

**3.)** In order to log into your EC2 virtual server using Terminal or PuTTY, you will need to create a Key Pair. Inside the EC2 dashboard under *NETWORK & SECURITY*, click on *Key Pairs* and then *Create Key Pair* > "name_your_key" > finish by clicking *Create*. (Figure 2)



Viewing: All Key Pairs

| Key Pair Name | Fingerprint |
|---|---|
| bas0_key | 69:41:a4:f4:b2:0d:20:86:1b:88:76:f6:78:3a:c9:71:34:ca:85:60 |

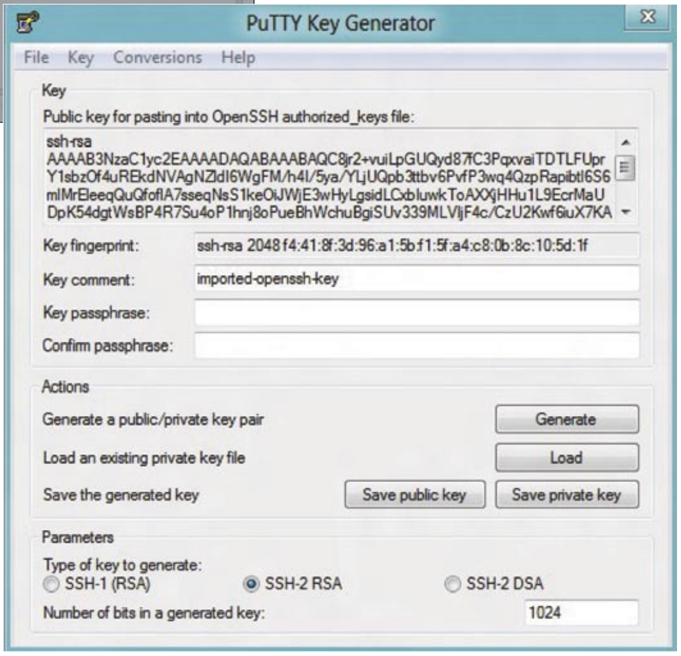
Create Key Pair

Key Pair Name: bsd0

0 Key Pairs selected

*Select a key pair above to view information about it here*

**Figure 2.** *Creating a Key Pair using PuTTYgen*



Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQC8jr2+vuiLpGUQyd87fC3PqxvaiTDTLFUpr
Y1sbzOf4uREkdNVAgNZldI6WgFM/h4l/5ya/YLjUQpb3ttbv6PvfP3wq4QzpRapibtI6S6
mIMrEleeqQuQfoflA7sseqNsS1keOiJWjE3wHyLgsidLCxbluwkToAXXjHHu1L9EcrMaU
DpK54dgtWsBP4R7Su4oP1hnj8oPueBhWchuBgiSUv339MLVljF4c/CzU2Kwf6iuX7KA
```

Key fingerprint: ssh-rsa 2048 f4:41:8f:3d:96:a1:5b:f1:5f:a4:c8:0b:8c:10:5d:1f

Key comment: imported-openssh-key

Key passphrase:

Confirm passphrase:

Generate a public/private key pair

Load an existing private key file

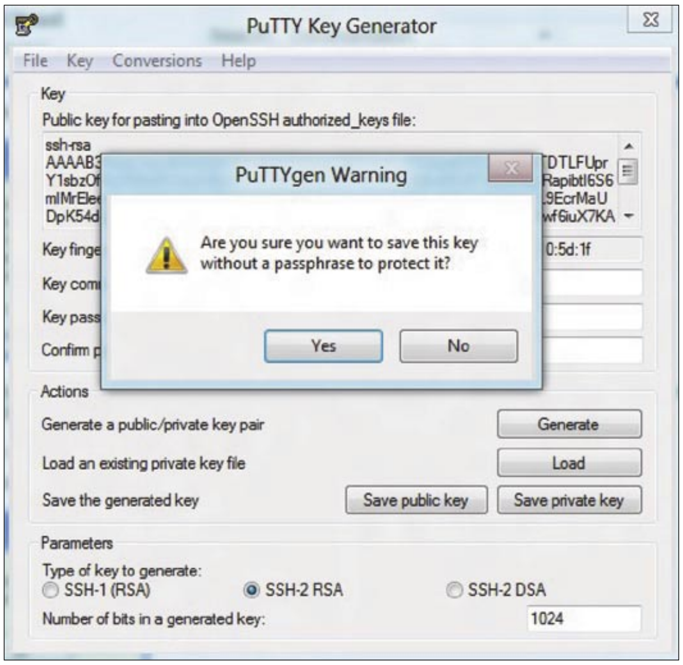
Save the generated key

Type of key to generate: SSH-1 (RSA) [✓ SSH-2 RSA] SSH-2 DSA

Number of bits in a generated key: 1024

**Figure 3.** *Creating a PuTTY compatible key pair*

**4.)** Next you will be prompted to download "your_key.pem". (Windows) Once downloaded run PuTTYGen.exe. From the *file* menu choose *load private key*. Open and load "your_key.pem". Once loaded (Figure 3) click on *save private key* button.



Are you sure you want to save this key without a passphrase to protect it?

Generate a public/private key pair

Load an existing private key file

Save the generated key

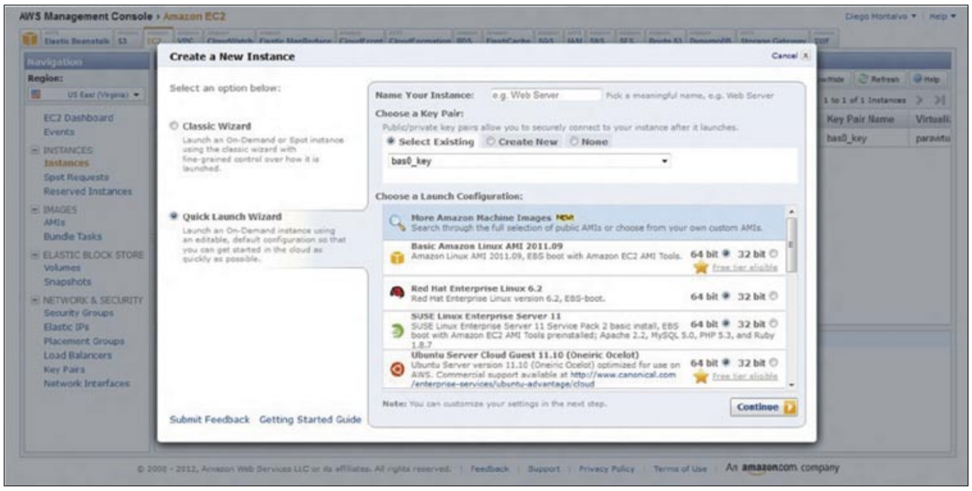
Type of key to generate: SSH-1 (RSA) [✓ SSH-2 RSA] SSH-2 DSA

Number of bits in a generated key: 1024

**Figure 4.** *Saving PuTTY key without a pass phrase*

**5.)** (Windows) When prompted *Are you sure you want to save this key without a passphrase to protect it*? Click *Yes* button (Figure 4). You will now have "your_key.ppk" file which can be used with PuTTY and the EC2 services.

**We are now ready to build a FreeBSD virtual server on EC2.**



Create a New Instance

Select an option below:

Classic Wizard
Launch an On-Demand or Spot instance using the classic wizard with fine-grained control over how it is launched.

Quick Launch Wizard
Launch an On-Demand instance using an editable, default configuration so that you can get started in the cloud as quickly as possible.

Name Your Instance: e.g. Web Server — Pick a meaningful name, e.g. Web Server

Choose a Key Pair:
Public/private key pairs allow you to securely connect to your instance after it launches.
[✓ Select Existing] Create New None
bas0_key

Choose a Launch Configuration:

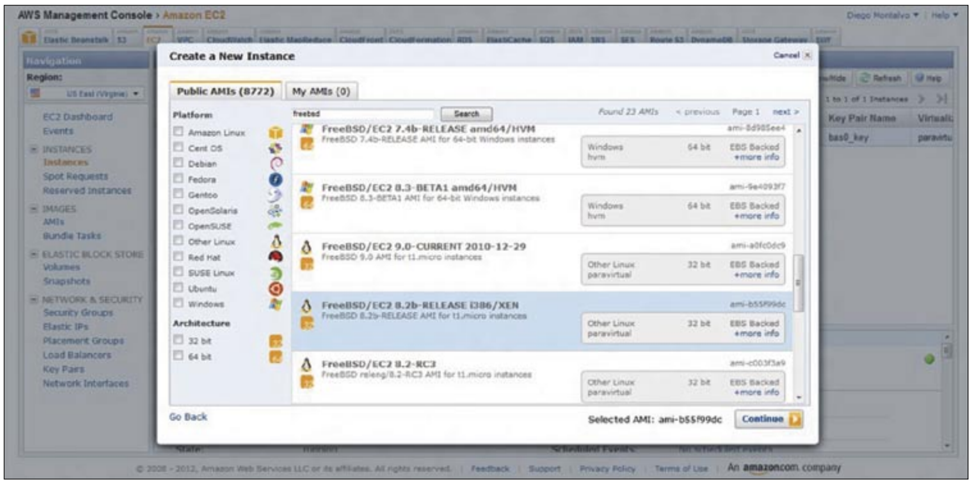
- More Amazon Machine Images NEW — Search through the full selection of public AMIs or choose from your own custom AMIs.
- Basic Amazon Linux AMI 2011.09 — Amazon Linux AMI 2011.09, EBS boot with Amazon EC2 AMI Tools. [✓ 64 bit] 32 bit — Free tier eligible
- Red Hat Enterprise Linux 6.2 — Red Hat Enterprise Linux version 6.2, EBS-boot. [✓ 64 bit] 32 bit
- SUSE Linux Enterprise Server 11 — SUSE Linux Enterprise Server 11 Service Pack 2 basic install, EBS boot with Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.0, PHP 5.3, and Ruby 1.8.7 [✓ 64 bit] 32 bit
- Ubuntu Server Cloud Guest 11.10 (Oneiric Ocelot) — Ubuntu Server version 11.10 (Oneiric Ocelot) optimized for use on AWS. Commercial support available at http://www.canonical.com/enterprise-services/ubuntu-advantage/cloud [✓ 64 bit] 32 bit — Free tier eligible

Note: You can customize your settings in the next step.

AWS Management Console › Amazon EC2
Submit Feedback
Getting Started Guide
Continue
Cancel
© 2008 - 2012, Amazon Web Services LLC or its affiliates. All rights reserved.

**Figure 5.** *Creating a new (AMI) virtual server*

**6.)** Inside the EC2 dashboard click on the *launch instance* button. Next *Name Your Instance* and under *Choose a Launch Configuration* click on *More Amazon Machine Images* then click *Continue*. (Figure 5)



Create a New Instance

Public AMIs (8772) | My AMIs (0)

Platform: Amazon Linux, Cent OS, Debian, Fedora, Gentoo, OpenSolaris, OpenSUSE, Other Linux, Red Hat, SUSE Linux, Ubuntu, Windows
Architecture: 32 bit, 64 bit

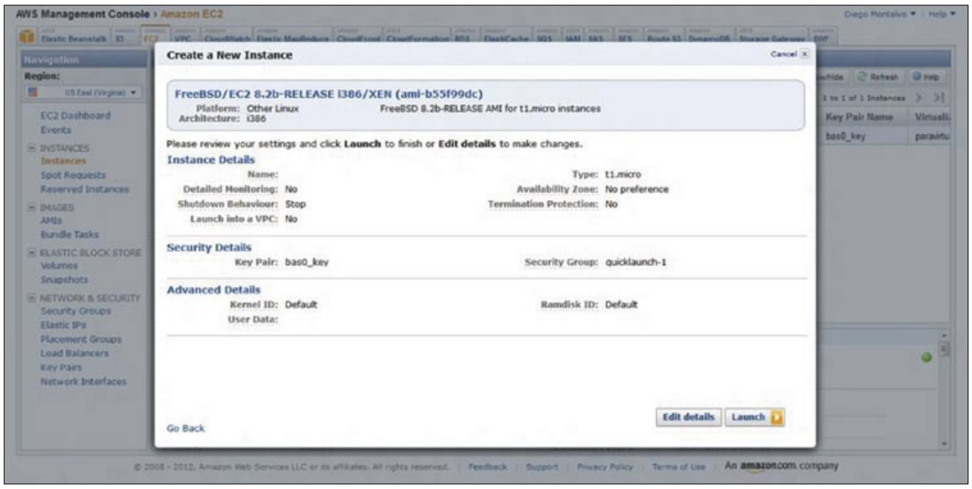
Search: freebsd — Found 23 AMIs — Page 1

| AMI | Description | ID | Platform | Arch | Root device |
|---|---|---|---|---|---|
| FreeBSD/EC2 7.4b-RELEASE amd64/HVM | FreeBSD 7.4b-RELEASE AMI for 64-bit Windows instances | ami-8d985ee4 | Windows hvm | 64 bit | EBS Backed |
| FreeBSD/EC2 8.3-BETA1 amd64/HVM | FreeBSD 8.3-BETA1 AMI for 64-bit Windows instances | ami-9e4093f7 | Windows hvm | 64 bit | EBS Backed |
| FreeBSD/EC2 9.0-CURRENT 2010-12-29 | FreeBSD 9.0 AMI for t1.micro instances | ami-a0fc0dc9 | Other Linux paravirtual | 32 bit | EBS Backed |
| FreeBSD/EC2 8.2b-RELEASE I386/XEN | FreeBSD 8.2b-RELEASE AMI for t1.micro instances | ami-b55f99dc | Other Linux paravirtual | 32 bit | EBS Backed |
| FreeBSD/EC2 8.2-RC3 | FreeBSD releng/8.2-RC3 AMI for t1.micro instances | ami-c003f3a9 | Other Linux paravirtual | 32 bit | EBS Backed |

Selected AMI: ami-b55f99dc

**Figure 6.** *Choosing a FreeBSD (AMI)*

**7.)** Type “freebsd” into the search box and click *Search*. All FreeBSD instances will be listed. In this tutorial we chose `FreeBSD/EC2 8.2b-RELEASE i386/XEN FreeBSD 8.2b-RELEASE AMI for t1.micro instances (ami-b55f99dc)`. Once you’ve chosen your Amazon Machine Image (AMI) click *Continue*. (Figure 6)



Create a New Instance

FreeBSD/EC2 8.2b-RELEASE i386/XEN (ami-b55f99dc)
Platform: Other Linux — FreeBSD 8.2b-RELEASE AMI for t1.micro instances
Architecture: i386

Please review your settings and click **Launch** to finish or **Edit details** to make changes.

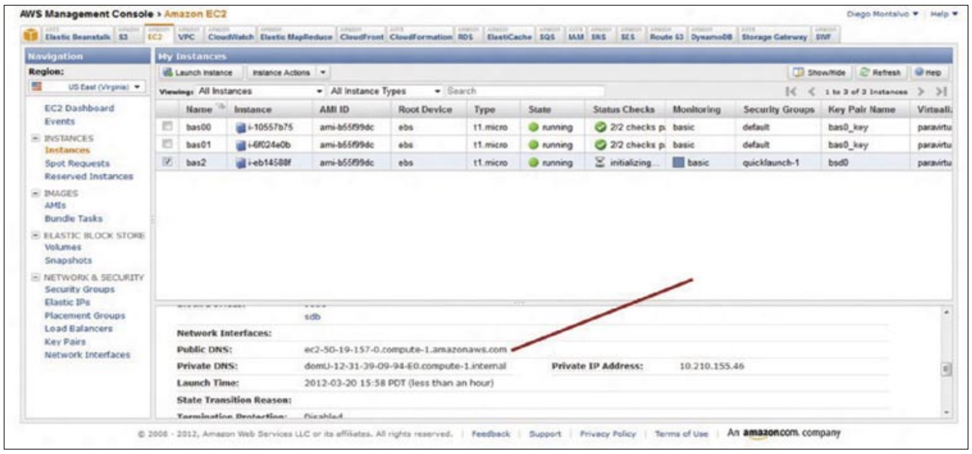
Instance Details
Name:
Type: t1.micro
Detailed Monitoring: No
Availability Zone: No preference
Shutdown Behaviour: Stop
Termination Protection: No
Launch into a VPC: No

Security Details
Key Pair: bas0_key
Security Group: quicklaunch-1

Advanced Details
Kernel ID: Default
Ramdisk ID: Default
User Data:

**Figure 7.** *Configuring and launching your new (AMI)*

**8.)** You may edit your virtual server settings by clicking on the *Edit Details. The default settings will suffice for this tutorial.* Click *Launch*. (Figure 7) then close the *Create a New Instance* window.



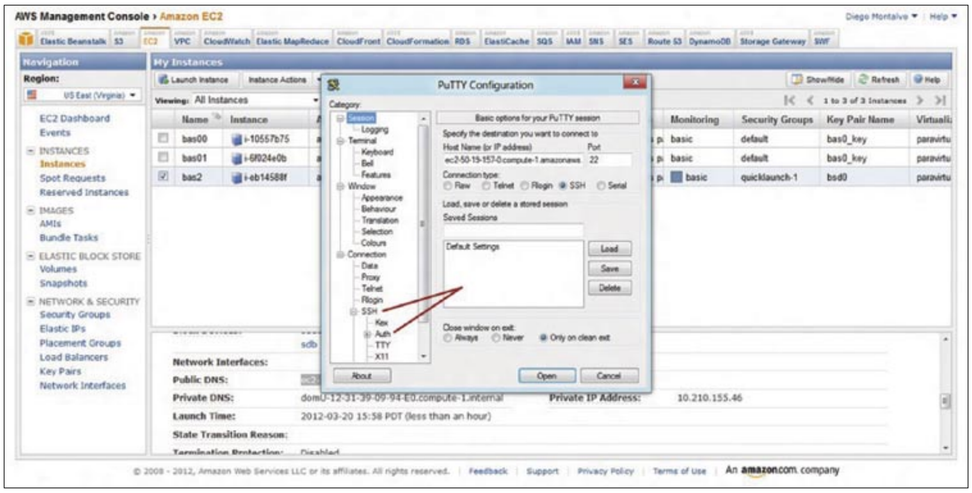
| | Name | Instance | AMI ID | Root Device | Type | State | Status Checks | Monitoring | Security Groups | Key Pair Name | Virtuali. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ☐ | bas00 | i-10557b75 | ami-b55f99dc | ebs | t1.micro | running | 2/2 checks p. | basic | default | bas0_key | paravirtu |
| ☐ | bas01 | i-6f024e0b | ami-b55f99dc | ebs | t1.micro | running | 2/2 checks p. | basic | default | bas0_key | paravirtu |
| ☑ | bas2 | i-eb14588f | ami-b55f99dc | ebs | t1.micro | running | initializing... | basic | quicklaunch-1 | bsd0 | paravirtu |

sdb
Network Interfaces:
Public DNS: ec2-50-19-157-0.compute-1.amazonaws.com
Private DNS: domU-12-31-39-09-94-E0.compute-1.internal  Private IP Address: 10.210.155.46
Launch Time: 2012-03-20 15:58 PDT (less than an hour)
State Transition Reason:
Termination Protection: Disabled

© 2008 - 2012, Amazon Web Services LLC or its affiliates. All rights reserved. | Feedback | Support | Privacy Policy | Terms of Use | An amazon.com company

**Figure 8.** *Copy and paste your Public DNS URL*

**9.)** Back inside the EC2 dashboard you will need to copy the *Public DNS* of your running instance for example: `ec2-1234-13.compute-1.amazonaws.com` so that you can connect via Terminal or PuTTY. (Figure 8)

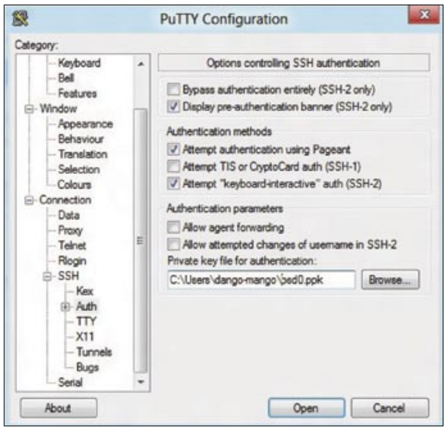


Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address): ec2-50-19-157-0.compute-1.amazonaws  Port: 22
Connection type: ( ) Raw ( ) Telnet ( ) Rlogin (•) SSH ( ) Serial
Load, save or delete a stored session
Saved Sessions
Default Settings
Close window on exit: ( ) Always ( ) Never (•) Only on clean exit

**Figure 9.** *Loading "your_key.ppk" into PuTTY*

**10.)** (Windows) Start PuTTY and in the *Host Name* textbox paste your Public DNS address. In the *Category* menu, under *Connection*, click *SSH*, and then *Auth*. The options controlling SSH authentication will be displayed. (Figure 9)

(PC-BSD) Locate „your_key.pem" file, next you will need to set permissions for your key. `# chmod 400 your_key.pem` (must not be publicly viewable to work)
(PC-BSD) `# ssh -i your_key.pem root@ec2-1234-13.your_aws.amazonaws.com`



Options controlling SSH authentication
☐ Bypass authentication entirely (SSH-2 only)
☑ Display pre-authentication banner (SSH-2 only)
Authentication methods
☑ Attempt authentication using Pageant
☐ Attempt TIS or CryptoCard auth (SSH-1)
☑ Attempt "keyboard-interactive" auth (SSH-2)
Authentication parameters
☐ Allow agent forwarding
☐ Allow attempted changes of username in SSH-2
Private key file for authentication:
C:\Users\dango-mango\bsd0.ppk

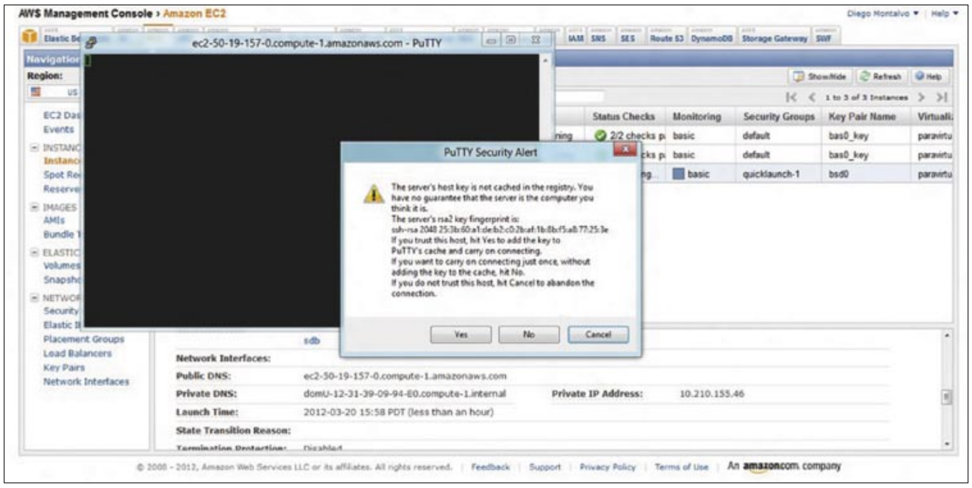
**Figure 10.** *Connecting to EC2 via PuTTY*

**11.)** (Windows) Click *Browse* and navigate to the PuTTY private key file you generated in the previous section. Click *Open*. (Figure 10)

(PC-BSD) You will see a response like the following:

```
The authenticity of host '
ec2-1234-13.your_aws.amazonaws.com (10.254.142.33)'
can't be established.
RSA key fingerprint is
00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00.
Are you sure you want to continue connecting (yes/no)?
```

Enter yes. You are now logged into your virtual server using PC-BSD. Happy Virtual Computing! (Figure 12)

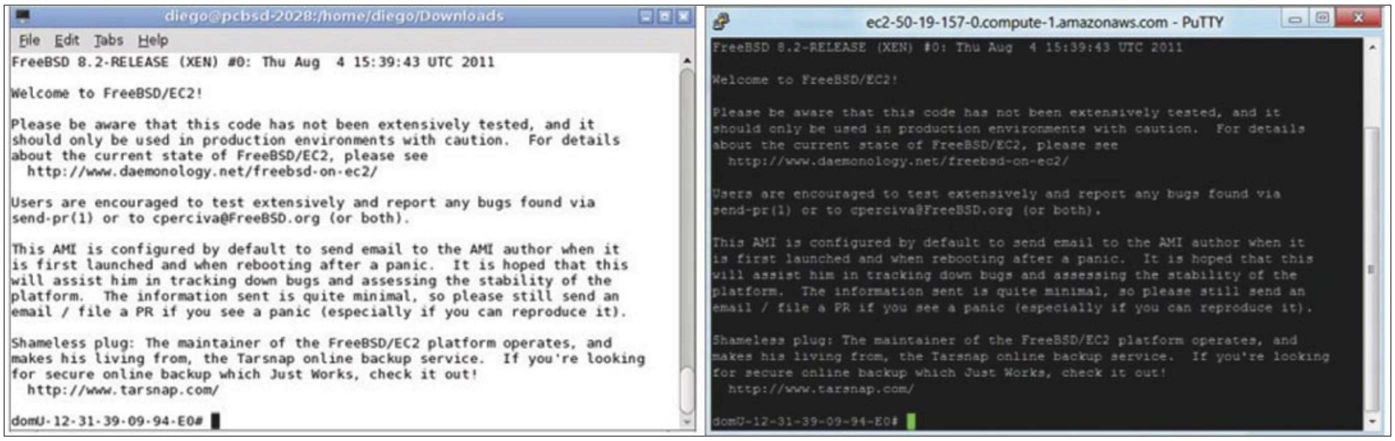

PuTTY Security Alert

The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.
The server's rsa2 key fingerprint is:
ssh-rsa 2048 25:3b:60:a1:de:b2:c0:2b:af:1b:8b:f5:a8:77:25:3e
If you trust this host, hit Yes to add the key to PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without adding the key to the cache, hit No.
If you do not trust this host, hit Cancel to abandon the connection.

Public DNS: ec2-50-19-157-0.compute-1.amazonaws.com
Private DNS: domU-12-31-39-09-94-E0.compute-1.internal
Private IP Address: 10.210.155.46
Launch Time: 2012-03-20 15:58 PDT (less than an hour)

**Figure 11.** *Connecting to EC2 via PuTTY*

**12.)** (Windows) You will be prompted with a PuTTY Security Alert Box – *The host key is not cached in the registry...* Click Yes If you trust the host connection (Figure 11). (Windows) Login as root and you are ready to install your favorite ports! (Figure 12)


```
FreeBSD 8.2-RELEASE (XEN) #0: Thu Aug  4 15:39:43 UTC 2011

Welcome to FreeBSD/EC2!

Please be aware that this code has not been extensively tested, and it
should only be used in production environments with caution.  For details
about the current state of FreeBSD/EC2, please see
  http://www.daemonology.net/freebsd-on-ec2/

Users are encouraged to test extensively and report any bugs found via
send-pr(1) or to cperciva@FreeBSD.org (or both).

This AMI is configured by default to send email to the AMI author when it
is first launched and when rebooting after a panic.  It is hoped that this
will assist him in tracking down bugs and assessing the stability of the
platform.  The information sent is quite minimal, so please still send an
email / file a PR if you see a panic (especially if you can reproduce it).

Shameless plug: The maintainer of the FreeBSD/EC2 platform operates, and
makes his living from, the Tarsnap online backup service.  If you're looking
for secure online backup which Just Works, check it out!
  http://www.tarsnap.com/

domU-12-31-39-09-94-E0#
```

**Figure 12.** *You are now connected to your virtual server*

**13.)** You are now connected to your virtual server on EC2. Cheers!

## References

- *http://aws.amazon.com/*
- *http://www.freebsd.org/*
- *http://www.chiark.greenend.org.uk/~sgtatham/putty/*
- *http://www.daemonology.net/blog/*

**DIEGO MONTALVO**

***Diego Montalvo is the founder of BuildaSearch.com a (site search engine web service) as well as Urloid.com a (URL shortening service). Diego by trade is a web developer and technical writer who enjoys the beach, socializing, staying up late, exercising and drinking an occasional pint. Feel free to contact Diego at diego@earthoid.com.***

RootBSD

PREMIERE VPS HOSTING

LATEST FREEBSD   FULL ROOT ACCESS

STARTING AT $20/MO   VPS AND DEDICATED

PRIVATE CLOUD

MULTIPLE DATACENTER LOCATIONS

FRIENDLY, KNOWLEDGEABLE SUPPORT STAFF

WWW.ROOTBSD.NET

# PostgreSQL: Replication

In the previous articles we saw how to set up a PostgreSQL cluster, how to manage backups (either logical or physical) and how internally transactions work. In this article we will see how it is possible to replicate a running cluster to another instance in order to have a fully mirrored and active "stand-by" node.

**What you will learn…**

- What is replication
- how to set up intra-cluster replication
- how to monitor a replica slave

**What you should know…**

- basic SQL concepts
- basic PostgreSQL concepts and configuration parameters
- basic shell commands

Most of the configuration will be done by simple shell scripts in order to both show required instructions and to allow readers to replay several time the experiments. Please note that configurations shown in this article are for didactic purposes only, and represent only a starting point for replica environments.

## Glance at Replication

Replicating a database cluster means setting up an on-line clone of such "master" cluster: the clone cluster must be kept up to date with the master one and must be available as soon as possible in the case the master node fails. Replication opens the door for *High Availability* (HA): you can replicate your main database on a slave (or more) to be sure that in the case the master node crashes or becomes unavailable the slave one can be promoted and substitute immediately the failed master.

Since the main idea behind replication is to have clones, it is strongly recommended to setup as much as possible similar environments, from the PostgreSQL instances to underlying operating systems and hardware. Ideally you can place replicating nodes wherever you want, starting from the same machine to different and far away systems. Of course the reliability of replication (and therefore of High Availability) is built on top of the reliability of the hardware, of the operating system, of the network connection, and so on. The configuration of the replication network depends on the aim you want to achieve, starting from a simple on-line backup node, a development environment, to a High Availability system (HA).

There are several replication solution for PostgreSQL, either OpenSource or commercial. This article will show only the natively supported replication methods available since the 9 series. It is worth noting that PostgreSQL replication is mature, since it comes out after years of development and testing, and is therefore production ready. It is also worth noting that PostgreSQL versions prior to 9 had external replication tools, and many of them still exist and are available to DBAs that have therefore a large set of choices depending on their needs, environment, and expertise.

## Setting up the Environment

In order to enable replication you must have at least one cluster with the role of "master" and one with the role of "stand-by". For the purposes of this article we will use instances on the same machine: `/postgresql/cluster1` is the master and is configured via rc variables: `/postgresql/clusterX` (being X a number) will be the stand-by instance running on the same machine but on different TCP/IP ports.

In order to allow readers to experiment with replication a couple of shell scripts has been built; such scripts configure and test a replicating standby node (identified by its number) running on the local machine. In particular the X cluster will be listening on a TCP/IP port increased by X with regard to the master listening one. In the following the details for each kind of replication will be discussed, but please consider that cluster numbering will have the following meaning:

- cluster 1 is the master node (listening on port 5432);
- cluster 3 is the log shipping replicated stand-by (listening on port 5435);
- cluster 4 is the log streaming replicated stand-by (listing on port 5436);
- cluster 5 is the Hot Standby replicated node (listening on port 5437).

Cluster number 2 is skipped because it was used in a former article to set up a Point In Time Recovery, and we don't want to mess the readers environment. Please also consider that for each replication example only the master (cluster 1) and the appropriate stand-by node will be active on the machine.

Please note that in this article we are going to use terms like "master" and "stand-by" to refer to the currently active cluster and the replicating one, even if in a replication environment such distinction is often useless, since a stand-by can be reconfigured to play as master and the master as stand-by and vice-versa. Moreover, readers will find the term "promote" to indicate a switch-over of a stand-by node, that is going to be promoted to the role of master due to a master crash or failure.

The script `01-createAndConfigureStandby.sh` (see Listing 4) creates the configuration of the stand-by node; it accepts the node number and the type to replication to configure. The script `00-workload.sh` (see Listing 5) provides a simple workload on the master and stand-by node to see how replication is applied. In the examples below the machine hosting all the cluster has an IP address of 192.168.200.2.

## A WAL to rule them all!

The solution PostgreSQL adopts to perform replication is elegant and efficient: each standby node is instrumented by the master to apply the same changes to the data so that, after a delay, the standby will contain the same data as the master. Such instrumentation is performed through the WAL logs (*Write Ahead Logs*), that contain a snapshot of the data and its changes performed by transactions. Sending the WALs to the standby nodes will allow each node to replay transactions done on the master to follow it during the data changes. The concept is really similar to the physical backup performed in the first article (see in particular PITR): a standby instance replays transactions done on the master using the WALs (similarly to what is done in case of a crash). The standby node continuously replays transactions, keeping updated with the master node.

Please note that each stand-by node must start from a physical copy of the master cluster, therefore each kind of replication must perform a `pg_start_backup()`, physical backup and `pg_stop_backup()` on the master before the replica can be started (see Listing 4).

## Log Shipping Replication

The first method explained here is "Log Shipping" replication. The configuration and the concept is really similar to the one behind the PITR discussed in the first article of this series. The master node archives the WALs over a location available to the standby node(s); the latter fetches the WALs and replays transactions continuously (see Figure 1). In this example the `/postgresql/pitr directory` is used as a common archive of the WALs, so the postgresql.conf file for the master node will include the following options to archive the WALs in the above directory:

```
wal_level='archive'
archive_mode=on
archive_command='cp -i %p /postgresql/pitr/%f'
```

The standby node will act as a pure clone, so it will not archive WALs by its own and therefore the `postgresql.conf` file contains:

```
wal_level='minimal'
archive_mode='off'
```

It is important to instrument the standby node that it is actually a standby and not a stand-alone master server, as well as where it can find archived WALs. To do this, a recovery.conf file is created:

```
standby_mode='on'
restore_command='cp /postgresql/pitr/%f %p'
archive_cleanup_command='pg_archivecleanup /postgresql/
                    pitr %r'
trigger_file='/postgresql/standby.3.trigger'
```

The first option (`standby_mode`) informs the cluster that it is acting as a standby node, and therefore it will be continuously replaying WALs transactions. The `restore_command` is used to fetch master's archived WALs. Of course this command must be the counterpart of the `archive _ command` specified in the master postgresql.conf and can be any command you want depending on where and how access to the WALs is provided. The `archive _ cleanup _ command` is an optional directive: it specifies a command that will be run over already replayed WALs; it is usually a command to remove old WALs to recover disk space. The `pg _ archivecleanup` command does exactly that, removing already processed (`%r`) WALs and removing them from the disk. It is worth noting that `pg _ archivecleanup` command does not know how many standby nodes are fetching WALs from the repository. In other words, imagine that more than one standby nodes are configured as stated above over the same WALs archive path: one standby will potentially erase WALs not yet processed by the other standby node. To solve this problem DBAs have to carefully configure the WALs archive storage, for example keeping a separate directory for each standby instance so that each instance can perform its own cleanup without damaging other standby node activities.

The last directive, `trigger_file`, is used for the standby node promotion. In fact, once a standby cluster is configured as above, it is running but it is not accepting user connections. In other words, trying to connect to the standby cluster will result in an error message:

```
$ psql -U bsdmag -p 5435 bsdmagdb
psql: FATAL: the database system is starting up
```

The database cluster is already started, but it is not ready yet, since it is continuously replaying WALs. Once the standby node finds the trigger file, it stops replaying WALs and become its own life as standalone cluster: from now on, the master node and the standby one are separated. The idea is that the triggering file can act as a notifier that something went bad on the master site and that the standby must substitute it. Readers can develop their own programs to query the master and, in case of failure, trigger the standby node. In the examples shown here promoting the standby node is done using the following command:

```
$ touch /postgresql/standby.3.trigger
```

Actions for performing a log shipping replications are summarized as follows:

- configure the master node to archive WALs;
- perform a physical backup of the master node in order to create the standby cluster;
- configure the standby cluster to not archive WALs (more on this later);
- create a `recovery.conf` file that will instrument the standby node to act as a standby and that contains a command to fetch (and possibly delete) WALs;
- when required, generate the triggering file to let the standby node to detach from the master and start accepting user connections.

In this example the standby node has been configured to not archive WALs by its own; this means that once the standby has been promoted and the master recovered

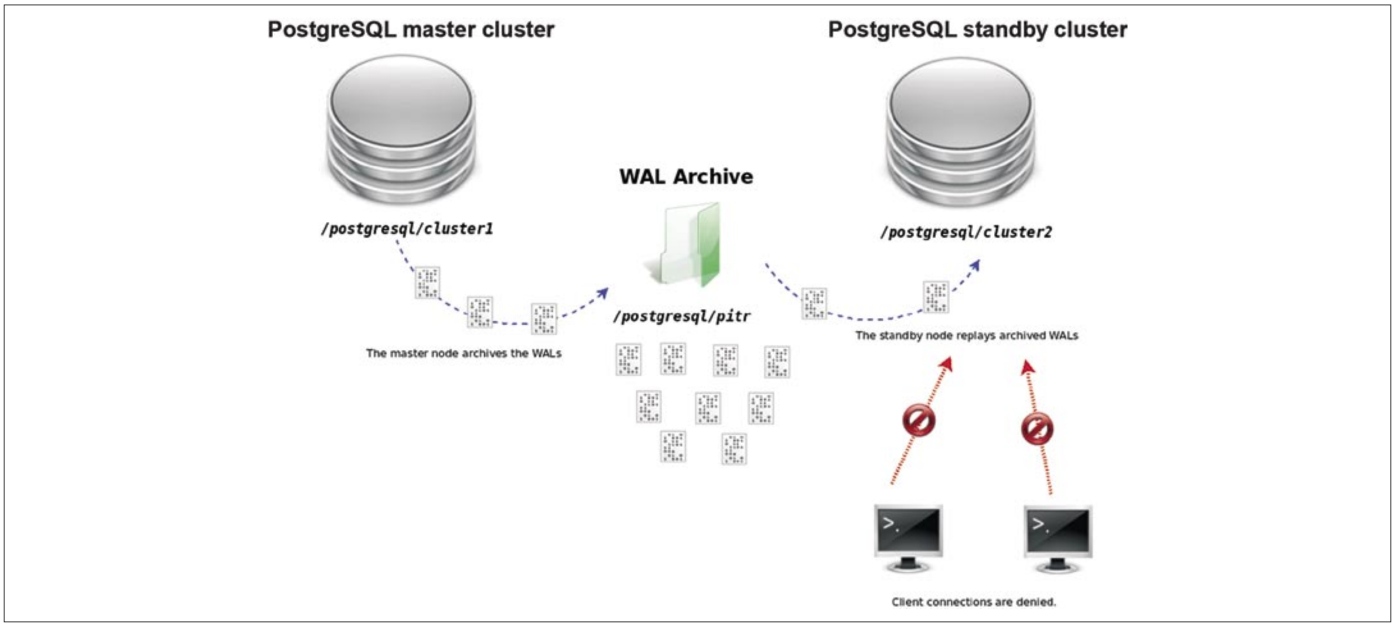

PostgreSQL master cluster
PostgreSQL standby cluster
WAL Archive
/postgresql/cluster1
/postgresql/cluster2
/postgresql/pitr
The master node archives the WALs
The standby node replays archived WALs
Client connections are denied.

**Figure 1.** *Scenario for log-shipping replication*

# BSDCAN 2012

THE BSD EVENT OF 2012

http://www.bsdcan.org/

*Ottawa, Canada*

*BSDCan 2012 – The event to be at this year*

BSDCAN 2012

The one, the only, BSDCan

## WHERE

Ottawa, Canada

## WHEN

09-10 May - tutorials
11-12 May - conference

## WHO

Contributors, developers, and users

## VENUE

University of Ottawa
http://www.uottawa.ca/

## AT FEES YOU CAN AFFORD

We plan to keep costs to a minimum. As such, the conference will be held at University of Ottawa and accommodation is available within the University residences. Hotels are also within close walking distance of the conference venue.

## WHAT DOES IT COST?

| Type | CAD |
|---|---|
| Individual | $195 |
| Corporate | $350 |
| Additional Corporate | $175 |
| Student | $60 |
| Tutorial (per half day) | $60 |

Comfortable accommodation is available on campus at very reasonable rates. See our website for details.

Take the BSDA Certification exam.
For details see
http://bsdcertification.org/

- An Introduction to Verifiedexec in NetBSD
- An Overview of Locking in the FreeBSD Kernel
- BSD Multiplicity
- auditdistd – Secure and reliable distribution of audit trail files
- Automated testing of libcurses in NetBSD
- FreeBSD on Microsoft Hyper-v
- BSDA Certification
- Crowdsourcing security
- Bullet Cache
- pfSense 2.1: IPv6 and more
- Fast reboots with kload
- Intro to DNSSEC
- NetBSD/mips
- Virtually-Networked FreeBSD Jails
- FreeBSD Unified Deployment and Configuration Management

from its failure, the two databases will remain not aligned. Depending on the environment you want to set up, the standby node could archive its own WALs so that the recovered master could be configured to play as standby; of course other scenarios are possible.

Listing 1 shows how to see log-shipping replication in action: the standby cluster is the number 3 and the workload will insert one million tuples in the *magazine* table and will generate a *test* table with 500 thousand tuples. The final result is that both clustres have the same data:

```
Activating the stand-by node...
==========================================
Tuples in the master node (magazine table)
1000000
Tuples in the master node (test table)
500000
==========================================
Tuples in the slave node (magazine table)
1000000
Tuples in the slave node (test table)
500000
==========================================
```

## Log Streaming Replication

In the log-shipping replication WALs were transferred from the master to the standby node(s) using an external tool and storage, in the example a shared directory and the `cp(1)` command. Log streaming replication on the other hand does not require a WALs shared archive, since each standby node can autonomously ask the master node for the WALs using a TCP/IP connection (see Figure 2). In this way there is no need for setting up a shared repository, which could be another point of failure in the whole system. Moreover, all the system is fully PostgreSQL based, since the connections are performed from a PostgreSQL instance to another instance directly. For security reasons, the master cluster requires a specific user with the `REPLICATION` grant to be used for incoming WAL requests. Standby nodes will connect to the master using such user (it is possible to share the same user or create a per-standby user). To create a replication user on the master node a command like the following must be issued:

```
CREATE USER replicator WITH REPLICATION LOGIN CONNECTION
LIMIT 1 PASSWORD 'replicatorPassword';
```

Beside the `REPLICATION` grant, the user must also be enabled for connection from the standby node, so the `pg _ hba.conf` file of the master node has to be modified with an entry that allows the standby node and the above user to connect for replication:

```
host replication replicator 192.168.200.0/24 trust
```

It is possible to restrict the Host Based Configuration to accept connections only from a single user instead of the entire network, as well as not trust (but ask for password); for the purposes of this example, the above configuration does suffice.

On the master cluster the configuration changes so that it has a set of dedicated processes to serve WALs to standby nodes; in this case there is no need for an `archive_command` to do a WAL archiving, so it is possible to place each command that will return a "true" status. The `postgresql.conf` file will contain therefore the following directives:

```
wal_level='archive'
archive_mode=on
archive_command='test 1 = 1'
archive_timeout=30
max_wal_senders=1
wal_keep_segments=50
```

In the above, a single process is dedicated on the master site to accept and serve incoming standby WAL requests. The `wal _ keep _ segments` parameter instruments the master to keep a number of WAL files (each 16 MB long) in the `pg _ xlog` directory just in case a stand-by connection fails. In this way, the stand-by node can lately fetch WAL required to the replication process. In other words, a stand-by connection can fail no more than the time required to serve `wal _ keep _ segments`, after that there is no guarantee the replication will be successful since the master could have already deleted a WAL needed by the stand-by. Please note that the `archive _ command` is anything can return a valid status (success) and that each 30 seconds a new WAL is forced to be written to disk (and therefore available to the standby nodes).

**Listing 1.** *Configuring and running a log shipping based replication node*

```
~> sh 01-createAndConfigureStandby.sh 3 logshipping
~> /usr/local/bin/pg_ctl -D /postgresql/cluster3 start
~> sh 00-workload.sh 3 promote
```

**Listing 2.** *Configuring and running a log streaming based replication node*

```
~> sh 01-createAndConfigureStandby.sh 4 logsstreaming
~> /usr/local/bin/pg_ctl -D /postgresql/cluster4 start
~> sh 00-workload.sh 4 promote
```

Similarly, the standby configuration does not require a `recovery_command` to restore the WALs from an archiving location; consequently there is no need for an `archive_cleanup_command` too. All the standby node needs now is the parameters for the connection back to the master node, so to ask to the latter the WALs. This is specified in the `recovery.conf` file as follows:

```
standby_mode='on'
primary_conninfo=' host=192.168.200.2 user=replicator'
trigger_file='/postgresql/standby.4.trigger'
```

As in the log shipping replication, the standby node has to be instrumented to play as a replica and not as a stand-alone cluster (`standby_mode='on'`), and it will start living on its own as soon as the `trigger_file` is found. As readers can see, the standby node will connect back to the master using the `primary_conninfo` properties, in particular the host IP address and the user for the connection.

It is possible to configure a log streaming replication environment running the commands of Listing 2, that perform the above steps and launch a workload on the master, promoting then the standby to a stand-alone. As in the previous case, the result of the workload is that both the master and the standby (now promoted) have the same number of tuples:

```
Activating the stand-by node...
==========================================
Tuples in the master node (magazine table)
1000000
Tuples in the master node (test table)
500000
==========================================
Tuples in the slave node (magazine table)
1000000
Tuples in the slave node (test table)
500000
==========================================
```

Actions for performing a log streaming replications are summarized as follows:

- configure the master node to archive WALs;
- perform a physical backup of the master node in order to create the standby cluster;
- configure the standby cluster to not archive WALs;
- create a `recovery.conf` file that will instrument the standby node to act as a standby and to ask the master for the WALs;
- when required, generate the triggering file to let the standby node to detach from the master and start accepting user connections. As readers can see,

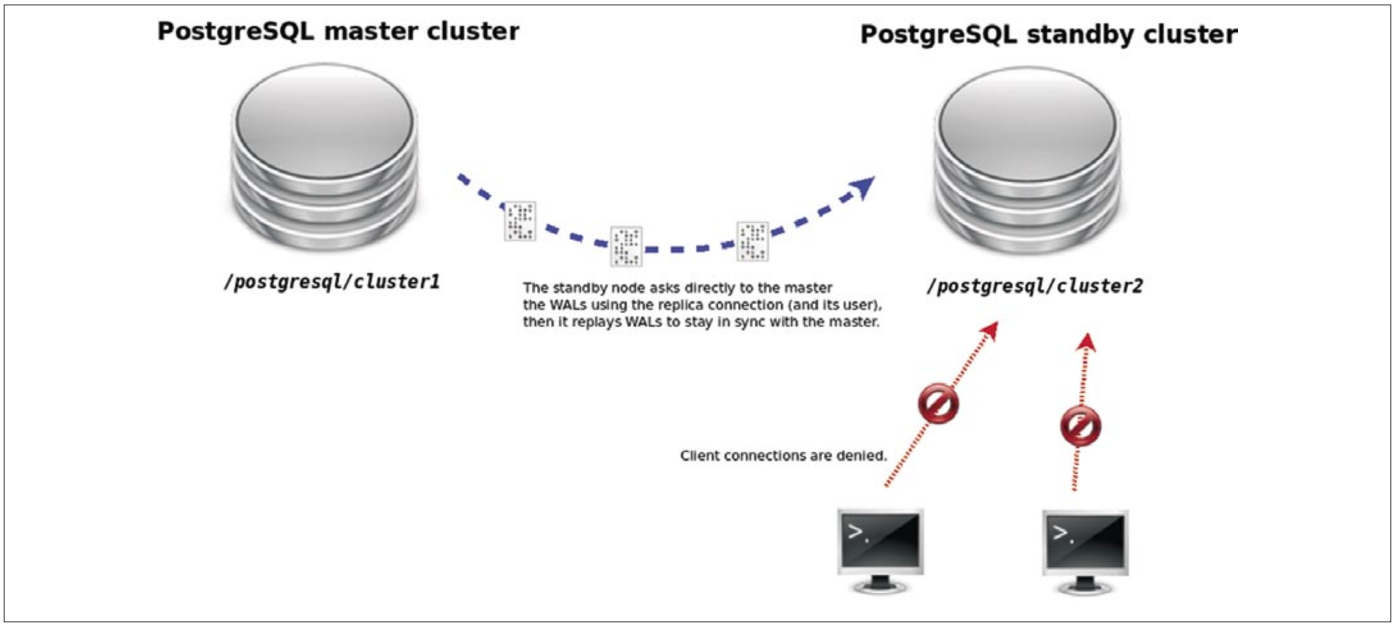

PostgreSQL master cluster
PostgreSQL standby cluster
/postgresql/cluster1
/postgresql/cluster2
The standby node asks directly to the master the WALs using the replica connection (and its user), then it replays WALs to stay in sync with the master.
Client connections are denied.

**Figure 2.** *Scenario for log-streaming replication*

**Listing 3.** *Configuring and running a streaming replication node*

```
~> sh 01-createAndConfigureStandby.sh 5 hotstandby
~> /usr/local/bin/pg_ctl -D /postgresql/cluster5 start
~> sh 00-workload.sh 5 show
```

the configuration procedure is almost the same of the previous case (log shipping), except for a few parameters.

## Making Stand-by nodes hot!

In the two previous replication scenarios both the stand-by nodes were forced to not accept incoming client connections until their promotion. PostgreSQL allows a so called "streaming replication", also known as "Hot Standby": in this scenario the stand-by nodes accept incoming client read transactions (e.g., `SELECT`) but do not allow for a client to modify the data. In other words, the stand-by node can be used as a more active node in the network architecture, for instance for load balancing purposes.

Configuration for Hot Standby is very similar to the one of the log streaming replication: the stand-by node asks for WAL logs to the master node, which in turn is informed to work in an Hot Standby configuration. Moreover, also the stand-by node knows to work in an Hot Standby configuration, and therefore accepts client connections and execute client statement in read-only transactions (see Figure 3). The following are the parameters required on the master `postgresql.conf` configuration file:

```
wal_level='hot_standby'
archive_mode=on
archive_command='test 1 = 1'
archive_timeout=30
max_wal_senders=1
replication_timeout=60
wal_keep_segments=16
```

Similarly to the case of log-streaming replication, the stand-by node will have a `recovery.conf` file that contains the working mode (Hot Standby) and where the master is (i.e., how to fetch WAL segments):

```
standby_mode='on'
primary_conninfo=' host=192.168.200.2 user=replicator'
trigger_file='/postgresql/standby.4.trigger'
```

Moreover, the stand-by node needs a special option in its `postgresql.conf` configuration file that informs the cluster to accept read-only connections:

```
hot_standby=on
```

It is interesting to see how the replica is going on: the script `00-workload.sh` can show the replication information if called with the *show* parameter as shown in Listing 3; in such case the output will be:

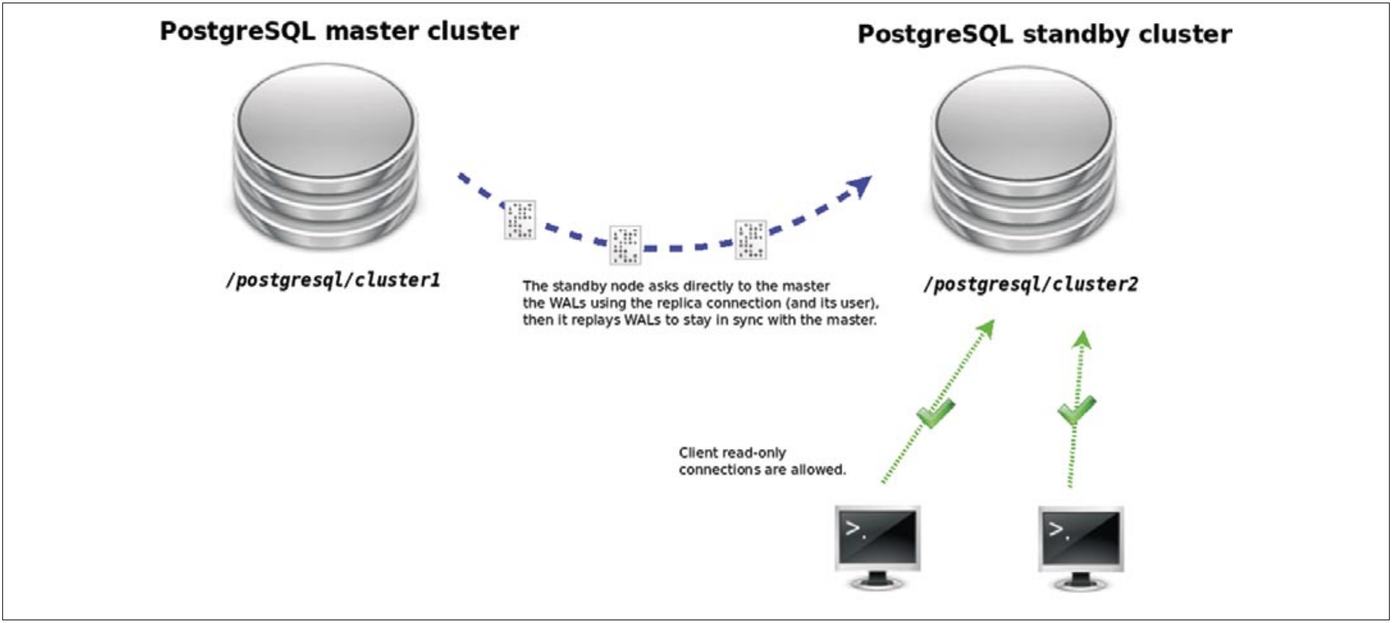

PostgreSQL master cluster
PostgreSQL standby cluster
/postgresql/cluster1
/postgresql/cluster2
The standby node asks directly to the master the WALs using the replica connection (and its user), then it replays WALs to stay in sync with the master.
Client read-only connections are allowed.

**Figure 3.** *Scenario for Hot Standby replication*

**Listing 4a.** *The script to create a replica configuration from scratch*

```
#!/bin/sh

STANDBY_CLUSTER_NUMBER=$1
REPLICATION_MODE=$2
REPLICATION_SYNC=$3
POSTGRESQL_ROOT=/postgresql
WAL_ARCHIVES=${POSTGRESQL_ROOT}/pitr
MASTER_CLUSTER=${POSTGRESQL_ROOT}/cluster1
DEST_CLUSTER=${POSTGRESQL_ROOT}/cluster${STANDBY_CLUSTER_
                   NUMBER}
RECOVERY_FILE=$DEST_CLUSTER/recovery.conf

HOST_IP=192.168.200.2
HOST_NET=192.168.200.0/24
REPLICATION_USER="replicator"
POSTGRESQL_CONF_TEMPLATE=/postgresql/
                   postgresql.conf.template


REPLICATION_MODE_LOGSHIPPING="logshipping"
REPLICATION_MODE_LOGSTREAMING="logstreaming"
REPLICATION_MODE_HOTSTANDBY="hotstandby"
REPLICATION_MODE_SYNC="sync"
REPLICATION_MODE_ASYNC="async"

# A function to set up the file system for the standby
                   node
# and to start the backup of the master cluster.
clone_master(){
# stop the cluster if running!
    /usr/local/bin/pg_ctl -D $DEST_CLUSTER stop >/dev/
                   null 2>&1
    sleep 2
    rm -rf $DEST_CLUSTER
    mkdir  $DEST_CLUSTER
    chown pgsql:pgsql $DEST_CLUSTER

    echo "Starting physical backup of the master node
                   [$BACKUP_LABEL]"
    psql -U pgsql -c "SELECT pg_start_backup('REPLICA-
                   $BACKUP_LABEL');" template1
    cp -R ${MASTER_CLUSTER}/* $DEST_CLUSTER
    rm -rf $DEST_CLUSTER/pg_xlog/* $DEST_CLUSTER/*.pid
                   $DEST_CLUSTER/recovery.* $DEST_
                   CLUSTER/*label*
    psql -U pgsql -c "SELECT pg_stop_backup();"
                   template1
    echo "Physical backup of the master node [$BACKUP_
                   LABEL] finished"

}

# Creates the recovery.conf file for the standby node in
                   the case
# of the log shipping.
create_recovery_file_for_log_shipping(){
    rm $TRIGGER_FILE > /dev/null 2>&1
    echo "standby_mode='on'" > $RECOVERY_FILE
    echo "restore_command='cp $WAL_ARCHIVES/%f %p'" >>
                   $RECOVERY_FILE
    echo "archive_cleanup_command='pg_archivecleanup
                   $WAL_ARCHIVES %r'" >> $RECOVERY_FILE
    echo "trigger_file='$TRIGGER_FILE'" >> $RECOVERY_FILE

}

# Creates a replication user on the master node to allow
                   standby to connect
# using such user to retrieve log WALs.
create_replication_user_on_master_if_not_exists(){
# check if the replication user exists and has the
                   replication capabilities
    REPLICATION_USER_EXISTS='psql -U pgsql -A -t -c
                   "SELECT rolreplication FROM pg_roles
                   WHERE rolname = '$REPLICATION_
                   USER';" template1'


    if [ -z "$REPLICATION_USER_EXISTS" ]
    then
       echo "Creating the replication user $REPLICATION_
                   USER"
        psql -U pgsql -c "CREATE USER $REPLICATION_USER
                   WITH REPLICATION LOGIN CONNECTION
                   LIMIT 1 PASSWORD '$REPLICATION_
                   USER';" template1

    else
       if [ $REPLICATION_USER_EXISTS = "t" ]
        then
          echo "Replication user $REPLICATION_USER
                   already exists on the master
                   cluster!"
        else
          if [ $REPLICATION_USER_EXISTS = "f" ]
            then
```

**Listing 4b.** *The script to create a replica configuration from scratch*

```
              echo "Replication user $REPLICATION_USER
                   already exists but does not have
                   replication capabilities! Altering
                   the role..."
                psql -U pgsql -c "ALTER ROLE
                   $REPLICATION_USER WITH REPLICATION;"
                   template1

            fi

        fi

}


# Adds an entry in the pg_hba.conf file to allow the
                   connection of the slave
# for the replication.
add_entry_pghba_if_not_exists(){
    grep $REPLICATION_USER ${MASTER_CLUSTER}/pg_hba.conf
                   > /dev/null 2>&1
    if [ $? -ne 0 ]
    then
       echo "adding an entry in the master pg_hba.conf"
        echo "host replication $REPLICATION_USER ${HOST_
                   NET} trust" >> ${MASTER_CLUSTER}/
                   pg_hba.conf
    fi

}

# A function to check that the configuration of the
                   master node is
# appropriate for the log shipping replication.
adjust_master_configuration_for_log_shipping(){
    local CONF=${MASTER_CLUSTER}/postgresql.conf
    cp $CONF /postgresql/postgresql.conf.beforeLogShippi
                   ng.$$ > /dev/null 2>&1

    # wal_level = 'archive'
    sed  -i .bak "s/wal_level[ \t]*=.*/wal_
                   level='archive'/g"        $CONF
    # archive_mode = on
    sed  -i .bak "s/archive_mode[ \t]*=.*/archive_
                   mode=on/g"        $CONF
    # archive command => copy to pitr directory
    sed  -i .bak "s,archive_command[ \t]*=.*,archive_
                   command='cp -i %p /postgresql/pitr/
                   %f',g"        $CONF

    # force a log segment every 30 seconds max
    sed  -i .bak "s/archive_timeout[ \t]*=.*/archive_
                   timeout=30/g"        $CONF
}
# A function to check that the configuration of the
                   master node is
# appropriate for the log streaming replication.
adjust_master_configuration_for_log_streaming(){
    local CONF=${MASTER_CLUSTER}/postgresql.conf
    cp $CONF /postgresql/postgresql.conf.beforeLogStream
                   ing.$$ > /dev/null 2>&1

    # wal_level = 'archive'
    sed  -i .bak "s/wal_level[ \t]*=.*/wal_
                   level='archive'/g"        $CONF
    # archive_mode = on
    sed  -i .bak "s/#*archive_mode[ \t]*=.*/archive_
                   mode=on/g"        $CONF
    sed  -i .bak "s,#*archive_command[ \t]*=.*,archive_
                   command='test 1 = 1',g"        $CONF
    # force a log segment every 30 seconds max
    sed  -i .bak "s/#*archive_timeout[ \t]*=.*/archive_
                   timeout=30/g"        $CONF
    # esnure at least one wal sender
    sed  -i .bak "s/#*max_wal_senders[ \t]*=.*/max_wal_
                   senders=1/g"        $CONF
    # log connections, so we can see how is connecting
                   to the master node
    sed  -i .bak "s/#*log_connections[ \t]*=.*/log_
                   connections=on/g"        $CONF

    sed  -i .bak "s/#*wal_keep_segments[ \t]*=.*/wal_
                   keep_segments=16/g"        $CONF

}



# A function to check that the configuration of the
                   master node is
# appropriate for the hotstandby replication.
adjust_master_configuration_for_hotstandby(){
    local CONF=${MASTER_CLUSTER}/postgresql.conf
    cp $CONF /postgresql/postgresql.conf.beforeHotStandb
```

SnB Qualità Informatica

Web solutions for your business

SnB - Viale Don Minzoni 5/B - Jesi (AN) - Italy
Tel: +39 0731204358 - http://www.snb.it - info@snb.it

**Listing 4c.** *The script to create a replica configuration from scratch*

```
                    y.$$ > /dev/null 2>&1

    # wal_level = 'archive'
    sed  -i .bak "s/wal_level[ \t]*=.*/wal_level='hot_
                    standby'/g"         $CONF
    # archive_mode = on
    sed  -i .bak "s/#*archive_mode[ \t]*=.*/archive_
                    mode=on/g"              $CONF
    sed  -i .bak "s,#*archive_command[ \t]*=.*,archive_
                    command='test 1 = 1',g"        $CONF
    # force a log segment every 30 seconds max
    sed  -i .bak "s/#*archive_timeout[ \t]*=.*/archive_
                    timeout=30/g"        $CONF
    # esnure at least one wal sender
    sed  -i .bak "s/#*max_wal_senders[ \t]*=.*/max_wal_
                    senders=1/g"         $CONF
    # log connections, so we can see how is connecting
                    to the master node
    sed  -i .bak "s/#*log_connections[ \t]*=.*/log_
                    connections=on/g"        $CONF
    # keep wal segments as emergency action
    sed  -i .bak "s/#*wal_keep_segments[ \t]*=.*/wal_
                    keep_segments=16/g"        $CONF

    # terminate the connections if the standby has
                    crashed
    sed  -i .bak "s/#*replication_timeout[ \t]*=.*/
                    replication_timeout=60/g"
                    $CONF
}
# Activate the master configuration for sync replication.
adjust_master_sync_replication(){
    local CONF=${MASTER_CLUSTER}/postgresql.conf
    SYNC_APP_NAME="sync_replication_$STANDBY_CLUSTER_
                    NUMBER"
    echo "Synchronous application name: $SYNC_APP_NAME"
    sed  -i .bak "s/#*synchronous_standby_names[ \
                    t]*=.*/synchronous_standby_
                    names=$SYNC_APP_NAME/g"        $CONF
    sed  -i .bak "s/#*synchronous_commit[ \t]*=.*/
                    synchronous_commit=on/g"        $CONF
    sed  -i .bak "s/#*replication_timeout[ \t]*=.*/
                    replication_timeout=1000/g"
                    $CONF
}
```

```
# Restart the master cluster.
restart_master_cluster(){
    echo "Restarting the master cluster..."
    /usr/local/etc/rc.d/postgresql restart
}


# Creates the recovery.conf file for the standby node in
                    the case

create_recovery_file_for_log_streaming(){
    rm $TRIGGER_FILE > /dev/null 2>&1
    echo "standby_mode='on'" > $RECOVERY_FILE
    echo "primary_conninfo=' host=$HOST_IP
                    user=$REPLICATION_USER'" >>
                    $RECOVERY_FILE
    echo "trigger_file='$TRIGGER_FILE'" >> $RECOVERY_FILE
}


create_recovery_file_for_log_streaming_sync_
                    replication(){
    rm $TRIGGER_FILE > /dev/null 2>&1
    echo "standby_mode='on'" > $RECOVERY_FILE
    echo "primary_conninfo=' host=$HOST_IP
                    user=$REPLICATION_USER application_
                    name=$SYNC_APP_NAME '" >> $RECOVERY_
                    FILE
    echo "trigger_file='$TRIGGER_FILE'" >> $RECOVERY_FILE
}




adjust_standby_configuration(){
    local CONF=$DEST_CLUSTER/postgresql.conf
    sed  -i .bak "s/#*port[ \t]*=[ \t]*\([0-9]*\)/
                    port=$DEST_PORT/g"   $CONF
    sed  -i .bak "s/wal_level[ \t]*=.*/wal_
                    level='minimal'/g"          $CONF
    sed  -i .bak "s/archive_mode[ \t]*=.*/archive_
                    mode='off'/g"        $CONF
    sed  -i .bak "s/max_wal_senders[ \t]*=.*/#max_wal_
                    senders=0/g"      $CONF
    sed  -i .bak "s/#*log_connections[ \t]*=.*/log_
                    connections=off/g"     $CONF
}
```

**Listing 4d.** *The script to create a replica configuration from scratch*

```
# A function to activate the hot standby for the standby
                node.
activate_hot_standby_on_standby_node(){
    local CONF=$DEST_CLUSTER/postgresql.conf
    sed  -i .bak "s/#*hot_standby[ \t]*=.*/hot_
                standby=on/g"    $CONF
}


activate_hot_standby_sync_on_standby_node(){
    local CONF=$DEST_CLUSTER/postgresql.conf
    sed  -i .bak "s/#*hot_standby[ \t]*=.*/hot_
                standby=on/g"    $CONF
    sed  -i .bak "s/#*wal_receiver_status_interval[
                \t]*=.*/wal_receiver_status_
                interval=500/g"    $CONF

}

# Print some final instructions for the usage of the
                standby.
print_final_info(){
    echo "Standby node $STANDBY_CLUSTER_NUMBER will
                listen on port $DEST_PORT"
    echo "Execute the following command to change the
                status of the standby"
    echo "in order to accept incoming connections:"
    echo
    echo "      touch $TRIGGER_FILE            "

    echo "To manage the cluster use:"
    echo
    echo "      /usr/local/bin/pg_ctl -D $DEST_CLUSTER
                {start | stop}"
    echo
    echo "To run a workload please execute"
    echo
    echo "      sh 00-workload.sh $STANDBY_CLUSTER_
                NUMBER [activate | show]"
}


# check the number of arguments
if [ $# -lt 2 ]
then
    echo "Usage:"
    echo "$0 <standby-number> <$REPLICATION_MODE_
                LOGSHIPPING | $REPLICATION_
                MODE_LOGSTREAMING |
                $REPLICATION_MODE_HOTSTANDBY>
                [start]"
    exit
fi

# check to operate on a cluster different from the
                master one
if [ $STANDBY_CLUSTER_NUMBER -eq 1 ]
then
    echo "Cluster #1 is the master!"
    exit
fi


# compute on which TCP/IP port the standby will be
                accepting connections
DEST_PORT='psql -U bsdmag -A -t -c "SELECT setting FROM
                pg_settings WHERE name = 'port';"
                template1'
DEST_PORT='expr $DEST_PORT + $STANDBY_CLUSTER_NUMBER'
echo "Destination port $DEST_PORT"

# where will be the recovery file for this standby node?
RECOVERY_FILE=$DEST_CLUSTER/recovery.conf
# which recovery file to use to activate the node?
TRIGGER_FILE=/postgresql/standby.${STANDBY_CLUSTER_
                NUMBER}.trigger


# which kind of replication should I do?
case $REPLICATION_MODE in
    ${REPLICATION_MODE_LOGSHIPPING})
    echo "Log shipping replication"
    BACKUP_LABEL=${REPLICATION_MODE_LOGSHIPPING}
    # 0) ensure the master has the right configuration
                for log shipping
    adjust_master_configuration_for_log_shipping
    restart_master_cluster

    # 1) clone the master into the standby directory
                filesystem
    clone_master

    create_recovery_file_for_log_shipping
    # 3) adjust the postgresql.conf file in the standby
                node
    adjust_standby_configuration
```

**Listing 4e.** *The script to create a replica configuration from scratch*

```
    ;;


    ${REPLICATION_MODE_LOGSTREAMING})
    echo "Log streaming replication"
    BACKUP_LABEL=${REPLICATION_MODE_LOGSTREAMING}
    # 0) ensure the master node has the right
                    configuration
    adjust_master_configuration_for_log_streaming
    restart_master_cluster

    # 1) clone the master into the standby directory
                    filesystem
    clone_master
    # 2) create the recovery.conf file
    create_recovery_file_for_log_streaming
    # 3) create the replication user on the master node
    create_replication_user_on_master_if_not_exists
    # 4) allow the standby node to connect back to the
                    master to allow for replication
    add_entry_pghba_if_not_exists
    # 5) set the standby configuration to not ship logs
    adjust_standby_configuration
    ;;


    ${REPLICATION_MODE_HOTSTANDBY})
    echo "Hot Standby replication"

    BACKUP_LABEL=${REPLICATION_MODE_HOTSTANDBY}
    # 0) ensure the master node has the right
                    configuration
    adjust_master_configuration_for_hotstandby
    restart_master_cluster


    # 1) clone the master into the standby directory
                    filesystem
    clone_master
    # 2) adjust parameters on master and create recovery
                    file on stand-by
    if [ "$REPLICATION_SYNC" = "$REPLICATION_MODE_SYNC"
                    ]
    then
       echo "Synchronous replication"
        adjust_master_sync_replication
        create_recovery_file_for_log_streaming_sync_
                    replication
    else
       create_recovery_file_for_log_streaming
    fi




    # 3) create the replication user on the master node
    create_replication_user_on_master_if_not_exists

    add_entry_pghba_if_not_exists
    # 5) set the standby configuration to not ship logs
    adjust_standby_configuration
    activate_hot_standby_on_standby_node
    if [ "$REPLICATION_SYNC" = "$REPLICATION_MODE_SYNC"
                    ]
    then
       restart_master_cluster
    fi
   ;;

    *)
        echo "Cannot proceed without the replication
                    method"
        exit
        ;;
esac


print_final_info


# shoudl I start the standby node?
if [ "$3" = "start" ]
then
   echo "Starting the standby node $DEST_CLUSTER"
    /usr/local/bin/pg_ctl -D $DEST_CLUSTER start
fi
```

```
Showing replica information
====== MASTER =======
1116 postgres: archiver process
=====================
====== STANDBY =======
1150 postgres: startup process
=====================
xlog location: master C9AF5F8 (9999598)
xlog location: standby C9AF5F8 (9999598)
Difference is 0
```

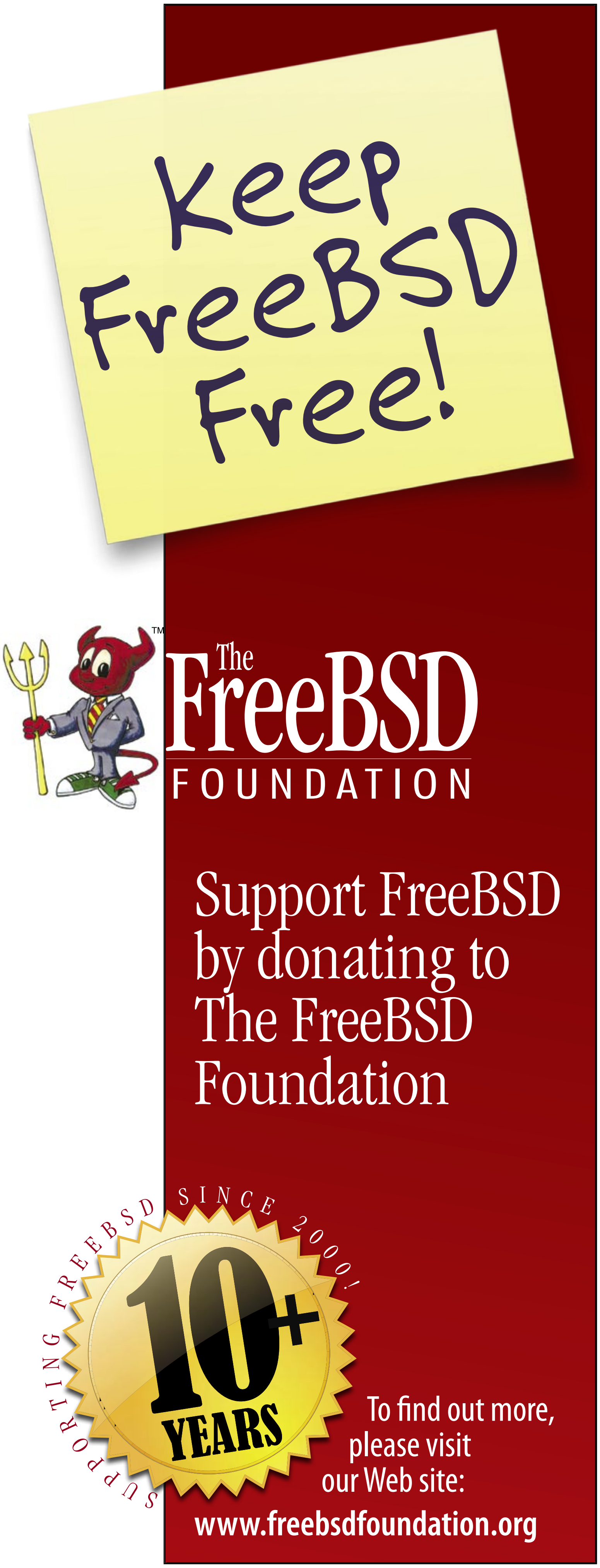
On the master node there is an "archiver" process that is in charge of keeping the WALs and serving them to the stand-by nodes; on the other hand there is a "startup" process that is replaying the WALs. It is possible to see the last WAL segment written by the master using the `pg_current_xlog_location()` function on the master node and `pg_last_xlog_replay_location()` on the stand-by. As shown in the above output, the difference between the segments is zero, meaning that the stand-by is aligned with the master node. The above script continuously prints the synchronizing information until the difference between the WAL segments is zero.

Actions for performing a Hot Standby replications are summarized as follows:

- configure the master node to archive the WALs for Hot Standby;
- perform a physical backup of the master node in order to create the standby cluster;
- configure the standby cluster to not archive WALs and to act as an Hot Standby node;
- create a `recovery.conf` file that will instrument the standby node to act as a standby and to ask the master for the WALs;

when required, generate the triggering file to let the standby node to detach from the master and start accepting user connections. Again, the configuration is almost the same as in the previous replication scenarios.

The Hot Standby described above is also called *asynchronous* replication, since the master node does not keep track of the stand-by status directly. In other words, once a transaction on the master is committed, the master immediately make the transaction persistent, even if the stand-by nodes have not yet required or replayed such transaction. What could happen now is that the connection between the master and the stand-by nodes goes down, making it impossible for the stand-bys to fully replicate


Keep FreeBSD Free!
The FreeBSD Foundation
Support FreeBSD by donating to The FreeBSD Foundation
SUPPORTING FREEBSD SINCE 2000!
10+ YEARS
To find out more, please visit our Web site:
www.freebsdfoundation.org

**Listing 5a.** *A workload script*

```
#/bin/sh

STANDBY_CLUSTER_NUMBER=$1
STANDBY_OPERATION=$2
POSTGRESQL_ROOT=/postgresql
WAL_ARCHIVES=${POSTGRESQL_ROOT}/pitr
MASTER_CLUSTER=${POSTGRESQL_ROOT}/cluster1
DEST_CLUSTER=${POSTGRESQL_ROOT}/cluster${STANDBY_CLUSTER_
                    NUMBER}
RECOVERY_FILE=$DEST_CLUSTER/recovery.conf
STANDBY_OPERATION_ACTIVATE="promote"
STANDBY_OPERATION_SHOWREPLICATION="show"


if [ $# -le 0 ]
then
   echo "Please specify the number of the cluster to
                    configure!"
    echo
    echo "Usage: $0 <cluster-number> [ $STANDBY_
                    OPERATION_ACTIVATE | $STANDBY_
                    OPERATION_SHOWREPLICATION ]"
    exit
fi

if [ $# -eq 1 ]
then
   STANDBY_OPERATION=$STANDBY_OPERATION_ACTIVATE
fi


if [ $STANDBY_CLUSTER_NUMBER -eq 1 ]
then
   echo "Cluster #1 is the master!"
    exit
fi


   DEST_PORT='psql -U bsdmag -A -t -c "SELECT setting
                    FROM pg_settings WHERE name =
                    'port';" template1'
    DEST_PORT='expr $DEST_PORT + $STANDBY_CLUSTER_
                    NUMBER'


echo "Operating mode is $STANDBY_OPERATION"
echo "The stand-by node $STANDBY_CLUSTER_NUMBER is
                    listening on $DEST_PORT"

TEST_TABLE_NAME="test$$"
COUNT_QUERY_MAGAZINE="SELECT count(*) FROM magazine;"
COUNT_QUERY_TEST="SELECT count(*) FROM $TEST_TABLE_
                    NAME;"

echo "Inserting tuples into the master node"
psql -U bsdmag -c "TRUNCATE TABLE magazine;" bsdmagdb
psql -U bsdmag -c "INSERT INTO magazine(id, title)
                    VALUES(generate_series(1,1000000),
                    'TEST-REPLICA');" bsdmagdb
echo "Tuples in the master node (magazine table)"
psql -U bsdmag -A -t -c "$COUNT_QUERY_MAGAZINE" bsdmagdb
psql -U bsdmag -c "CREATE TABLE $TEST_TABLE_NAME(pk
                    serial NOT NULL, description text);"
                    bsdmagdb
psql -U bsdmag -A -t -c "INSERT INTO $TEST_
                    TABLE_NAME(pk, description)
                    VALUES(generate_series(1,500000),
                    'NEW-TABLE-TEST');" bsdmagdb


if [ "$STANDBY_OPERATION" = "$STANDBY_OPERATION_
                    ACTIVATE" ]
then
   echo "Activating the stand-by node..."
    sleep 30
    touch /postgresql/standby.${STANDBY_CLUSTER_
                    NUMBER}.trigger
    sleep 10
    echo "==========================================="
    echo "Tuples in the master node (magazine table)"
    psql -U bsdmag -A -t -c "$COUNT_QUERY_MAGAZINE"
                    bsdmagdb
    echo "Tuples in the master node (test table)"
    psql -U bsdmag -A -t -c "$COUNT_QUERY_TEST" bsdmagdb

    echo "Tuples in the slave node (magazine table)"
    psql -U bsdmag -A -t -p $DEST_PORT -c "$COUNT_QUERY_
                    MAGAZINE" bsdmagdb
    echo "Tuples in the slave node (test table)"
    psql -U bsdmag -A -t -p $DEST_PORT -c "$COUNT_QUERY_
                    TEST" bsdmagdb
    echo "==========================================="
else
   if [ "$STANDBY_OPERATION" = "$STANDBY_OPERATION_
                    SHOWREPLICATION" ]
    then
```

**Listing 5b.** *A workload script*

```
      WAL_DIFFERENCE=1
       while [ $WAL_DIFFERENCE -gt 0 ]
       do

          echo "Showing replica information"
            # show log shipping processes
           echo "====== MASTER ======="
           pgrep -f -l -i archiver
           pgrep -f -l -i postgres | grep sender
           echo "====================="
           echo "====== STANDBY ======="
           pgrep -f -l -i startup
           pgrep -f -l -i postgres | grep receiver
           echo "====================="


           MASTER_XLOG_LOCATION='psql -U pgsql -A
                   -t -c "SELECT pg_current_xlog_
                   location();" template1 | sed
                   's|[0-9]/\([0-9]\)*||g''
          STANDBY_XLOG_LOCATION='psql -U pgsql -p
                   $DEST_PORT -A -t -c "SELECT pg_last_
                   xlog_replay_location();" template1 |
                   sed 's|[0-9]/\([0-9]\)*||g''


           obase=10
           ibase=16
           export obase
           export ibase
           MASTER_XLOG_LOCATION_10='echo $MASTER_
                   XLOG_LOCATION | bc'
           STANDBY_XLOG_LOCATION_10='echo $STANDBY_
                   XLOG_LOCATION | bc'
          echo "xlog location: master  $MASTER_XLOG_
                   LOCATION  ($MASTER_XLOG_LOCATION_10)"
           echo "xlog location: standby $STANDBY_
                   XLOG_LOCATION ($STANDBY_XLOG_
                   LOCATION_10)"
           WAL_DIFFERENCE='expr $MASTER_XLOG_
                   LOCATION_10 - $STANDBY_XLOG_
                   LOCATION_10'
           echo "Difference is $WAL_DIFFERENCE"
       done


    fi
fi
```

the master status. To avoid this problem, starting from the 9.1 release, PostgreSQL supports also the *synchronous* replication: each time a transaction is committed on the master, the master waits an acknowledge from the stand-by nodes indicating they have also applied such transaction. As readers can imagine, having a master node waiting indefinitely for a stand-by acknowledge can quickly become a failure (e.g., if the stand-by nodes are not able to send back their acknowledge), and therefore the master node will wait only for a specified amount of time, after that will apply anyway the transaction commit.

The synchronous replication is configured using a list of stand-by names the master will be informed are replicating it. Only one of multiple stand-by nodes will be queried for acknowledge commits by the master, and other stand-by will act as synchronous stand-by nodes once the former stop responding to the master. In the master `postgresql.conf` file the following parameters must be configured:

```
synchronous_standby_names=sync_replication_6
replication_timeout=1000
synchronous_commit=on
```

in such case the stand-by node will be named `sync_replication_6`. The `replication_timeout` parameter allows the master to discard a replication connection that has been inactive for the specified number of milliseconds. In the `recovery.conf` of each stand-by such name must be repeated:

```
standby_mode='on'
primary_conninfo=' host=192.168.200.2 user=replicator
                   application_name=sync_replication_6 '
trigger_file='/postgresql/standby.6.trigger'
```

Finally, in the stand-by configuration it is appropriate to place a configuration parameter like the following:

```
wal_receiver_status_interval = 500
```

in order to send back to the master information about the replication status of the stand-by. This is useful to avoid the master to drop the replication connection too early in the case of a network delay.

The above configuration can be obtained again using the script of Listing 4 and passing the last argument `sync` as shown below:

```
~> sh 01-createAndConfigureStandby.sh 5 hotstandby sync
~> /usr/local/bin/pg_ctl -D /postgresql/cluster5 start
```

## On The Web

- PostgreSQL official Web Site: *http://www.postgresql.org*
- ITPUG official Web Site: http://www.itpug.org
- PostgreSQL Replication Documentation: *http://www.postgresql.org/docs/9.1/interactive/high-availability.html*
- PostgreSQL Streaming Replication Wiki: *http://wiki.postgresql.org/wiki/Streaming_Replication*
- Replication solutions comparison: *http://wiki.postgresql.org/wiki/Replication,_Clustering,_and_Connection_Pooling*
- Londiste Tutorial: *http://wiki.postgresql.org/wiki/Londiste_Tutorial*

If now the master node cannot be sure of a replicated transaction on the stand-by, a message similar to the following one will be reported in the logs:

```
The transaction has already committed locally, but might
not have been replicated to the standby.
```

This means that there is no absolute guarantee that the stand-by is still replicating the master node.

### General Considerations

Log streaming replication and Hot Standby are really similar from an operative point of view, and therefore some considerations related to their configuration can be common. The first one is about the WAL archiving performed by the master node: the WAL segments must be kept enough time to allow the stand-by to ask them again in the case of a network failure. Therefore the parameter `wal_keep_segments` must be set accordingly, considering also the disk space available and the fact that each segment will occupy 16 MB. A second consideration is about the `max_wal_senders` parameter of the master node, that must be at least the number of stand-by nodes that are supposed to be replicating the data. Such parameter in fact provides the number of processes that are going to serve WAL requests by the stand-by node(s).

If something goes wrong between the master and the stand-by node, for instance the master stops responding due to a network problem, the slave will notify this in the logs:

```
FATAL:  replication terminated by primary server
FATAL:  could not connect to the primary server:
could not connect to server: Connection refused
```

If now the master node becomes available again, the stand-by node starts receiving again WALs and therefore will become up to date again. Nevertheless, it is worth noting how much it is important to monitor the status of the replicating clusters to ensure that everything is working fine.

Please take into account that in streaming replication, either synchronous or asynchronous, the master node will not honour a normal shutdown request until all the WAL segments have been served to the attached stand-by nodes.

As already stated, transaction executed on hot stand-by nodes must be read-only, in particular no transaction id (`xid`) will be assigned to transactions started on the stand-by nodes, as well as such transactions will be unable to write data to the "local" WAL.

## Summary and Coming Next

Thanks to the replication capabilities of PostgreSQL setting up a mirrored environment is straightforward and allows DBAs to provide an environment for High Availability. In the next article we step back to SQL to see some of the features available to DBAs and developers.

**LUCA FERRARI**

***Luca Ferrari lives in Italy with his wife and son. He is an Adjunct Professor at Nipissing University, Canada, a co-founder and the vice-president of the Italian PostgreSQL Users' Group (ITPUG). He simply loves the Open Source culture and refuses to log-in to non-Unix systems. He can be reached on line at http://fluca1978.blogspot.com***

CENTRAL EUROPEAN

# BSD-Day

2012

with BSDA Exam

BSD CERTIFICATION.ORG

University of Applied Sciences Technikum Wien

http://bsdday.eu/2012

May 5

# Interview with Mark Price, President of Tranquil Hosting, owner of RootBSD

Technology entrepreneur, proven experience working with and integrating a wide range of systems and networks. His specialties are: server management, advanced networking, web hosting, Linux, BSD, open source software.

## INTO

### Can you tell our readers how it happened that you are where you are and you do what you do?

**MP:** As a kid, I loved to play with computers, as most readers of BSDmag can probably relate to. I first explored both Linux (Slackware) and BSD by installing FreeBSD. Back then, you had to really want to try a new OS since it required ordering a CD-ROM or spending your nights downloading files from Walnut Creek FTP site on a slow dialup connection. As the growth of the web exploded, I've been putting my interest in Unix to work by building hosting services for our growing customer base.

### How it happened that RootBSD was created?

**MP:** As you know, most mass-market web hosting services are built on Linux. BSD users seem to be a more 'do it yourself' type who don't necessarily just want to just run some automated installers and be like everyone else.

Due to our interest in BSD we were just curiously searching for BSD hosts to see how BSD hosts were addressing the technical challenges of offering *Virtual Private Servers* (VPS), since a lot of the mainstream technology was made specifically for Linux. We were surprised to find that there weren't any popular BSD VPS options out there, so we decided to create one. It was mainly just a self-serving project, as we didn't think many BSD users would want to sign up for such a service anyway.

### Why clouds? Why hosting solutions? Is there a greater idea behind it or it was just a coincident?

**MP:** Cloud hosting solutions are the most affordable for consumers. The cost of a comparable dedicated server is typically much greater than that of a virtualized server.

Private Clouds: The amount of resources required by modern services can easily exceed that of a single larger server. Clouds allow for services to scale up past the limitations of...

### What kind of technical difficulties and problems you had to go through?

**MP:** Specifically with developing our own BSD hosting service, we've had to deal with lots of little technical problems. At first it was with tuning and tweaking the jail code in FreeBSD. Later on it was with doing some complex networking configurations in Xen. And of course, we've also been through many different iterations of ways to deploy FreeBSD installations with automated tools while still addressing customer's specific needs. These technical challenges are things that I, and the rest of the RootBSD staff, really enjoy.

## DEFINITION OF CLOUD

### What is cloud that we didn't have before?

**MP:** I don't think cloud is anything specifically new. I think its just a term that describes the trend in businesses outsourcing more IT functions. My experience with IT people in general is that IT people are very possessive and territorial, wanting to have lots of servers doing lots of things in-house. The 'cloud' idea just says "OK, we are going to outsource some of this boring technology stuff and instead concentrate on what's really important to us".

### Is cloud in BSD something different than in other systems?

**MP:** I think that cloud means something slightly different to each person, as well as to each developer and software architect. It is fascinating to see how BSD developers and users are building cloud solutions, and in many ways, the cloud is blind to specific operating systems. For example, Colin Percival has figured out how to run FreeBSD on Amazon's large EC2 cloud, even though Amazon doesn't offer FreeBSD support themselves.

### What are the advantages of cloud for BSD users?

**MP:** Thinking of cloud can bring advantages to all BSD users and IT professionals. The days of spending days setting up and configuring a new server have been replaced by a myriad of rapid deployment technologies and all sorts of on-demand technology services available now that we couldn't imagine 15 years ago.

## IMPLEMENTATION

### Could you give us some samples and case studies where cloud has been and is being implemented using BSD?

**MP:** Yes, there has been a lot of exciting development in BSD that ties in nicely to cloud concepts:

Sun's ZFS filesystem has been implemented in FreeBSD and is now incredibly stable and robust. ZFS offers all sorts of flexible ways to think about storage rather than being constrained to just legacy hardware solutions such as RAID cards which are designed for fixed storage configurations. ZFS is useful for scaling storage needs and also replicating storage. Jails have been used for years on FreeBSD hosts to segment applications into containers. The jail environment can make it easy for developers and administrators to easily "spin up" and "spin down" test environments without having to do traditional OS installation, create filesystems, etc. CARP is a great networking technology to allow multiple hosts to share an IP, an excellent example of using multiple hosts to increase overall availability. The CARP protocol was developed by the OpenBSD team and is now available in other BSD flavors.

### Can you tell us step by step how to build affordable and stable clouds using BSD?

**MP:** There is no such thing as a "one-size-fits-all" solution, but I would recommend looking at the jail functionality in FreeBSD to learn how to rapidly setup and take down jails. Taking this a step further, one can use ZFS to provide fault-tolerant storage for a cloud of FreeBSD jails. This is something that anyone can build at home with just a few older PCs.

## SECURITY

### Where are weak points of security in cloud?

**MP:** In a very broad perspective, one of the weak points is just in first deliniating where the security boundaries are. If you make use of cloud providers for running virtual servers and storage, what security steps is the provider taking for you? What is the customer's responsibility to secure? I'd say this is the biggest weak point, the easy of use of some cloud technologies makes it easy to not think about security implications.

### How do your company handle security when dealing with the cloud?

**MP:** We're always looking for ways to improve on security. We provide our new VPS customers with some general tips on how to keep up with securing their systems. From time to time, we will scan our whole network for certain popular vulnerabilities if we suspect a particular issue may be affecting customers. Taking it a step further, we offer custom solutions such as private clouds for businesses where we maintain and manage their environment with dedicated firewalls.

### How do you verify rumors/or not for 0-day exploits and such?

**MP:** Hopefully, the systems have been setup from beginning to not allow any unneeded access or possible attack vectors. This is done by turning off unneeded daemons (fortunately FreeBSD is very good out of the box in this respect!) and closing off network ports with a firewall. When an exploit is uncovered, we think that the FreeBSD security team does a great job with evaluating the risk of potential vulnerabilities, and addressing them quickly.

If you're not already on the FreeBSD security mailing list, read more here: *http://security.freebsd.org/.*

## FUTURE

### What will be the future of cloud in BSD in your opinion?

**MP:** Within the next few years the "cloud" will become ubiquitous across all platforms, including BSD. While there will likely always be a need for some internal or onsite deployment of resources, as time goes on we will see an even larger shift towards flexible hosted infrastructures. Components of the "cloud" such as virtualization, SaaS, and distributed storage are all quickly becoming the standard as opposed to the cutting edge. The benefits of cloud infrastructure far outweigh any potential pitfalls, so it's an inevitable conclusion.

# The Greater Benefits of Open Source Software

When Patrycja contacted me about contributing an article for the April 2012 issue, I didn't have any material ready for print. Yes, I have a few ideas but they remain works-in-progress at the moment. Instead, I opted to write on what I think are the benefits of Open Source software. I did have difficulty finding statistics on BSD operating systems, and used Apache as an example.

In contrast to proprietary software produced by most commercial manufacturers, Open Source software is written and perfected by volunteers, who freely share the programming code that would otherwise be kept secret.

Under the terms of the most popular Open Source licenses, anyone can redistributed the software without paying fees to its author, and anyone can modify it if they distributed the new version under the original terms – Open Source and non-proprietary. In recent years, Open Source products such as the BSD family of operating systems (and Linux) have emerged as significant competitors to proprietary software products like Microsoft Windows.

The rise of Open Source software and the challenge it poses to proprietary software producers is just the latest example of a long-standing tension between open and closed systems in the configuration of the global information infrastructure. Because the configuration of the information infrastructure has a strong impact on both the global economy and world politics, the rise of Open Source software is something that is best seen in a broader light, giving consideration to implications beyond the software industry. This phenomenon of Open Source software actually has quite a lot to teach us about some very general things – about the political economy of the Internet era, and actually, just as important, about the social process by which we are coming to understand it and the kind of policy process that is going to grow up around it.

Let's address Open Source as a market phenomenon, stating some of the basic facts and seeking to clarify some misconceptions that have emerged in recent treatment of the issue.

First, Open Source software movement, per se, has barely begun. The method of collaboration involved in producing Open Source software dates back to the 1950s and 1960s, but its manifestation in the market today, and the legal and policy issues that accompany it, are both new.

Second, discussions of Open Source have combined elements of truth with a lot of hype and misinformation. Open Source does not represent a "revolution in capitalism" or anything of the sort, and with proclamations like these it is easy to lose sight of what really is unique about the movement. To make matters worse, Open Source is sometimes actively misrepresented by stakeholders who feel threatened by its rise. One widespread misconception that has resulted is that Open Source software is inherently *free* software – a commodity that must be given away without charge and cannot be leveraged into a successful business model. Open Source software is free in the sense of freedom to view and modify its source code – not in the sense of zero price. For-profit companies such as BSD Virtual Machines have found ways to make Open Source software available to people by adding value in the form of customer service and support.

While proprietary software still dominates the market for personal computers (PCs), Open Source products are used widely on the servers that power the Internet – a development with profound implications for the software industry and Internet economy. As of March 2012, 65.2% of Web servers were running Apache, while only 13.8% ran Microsoft (according to Netcraft). Market share for the Open Source Apache server software is substantially higher than for any competing product, and Apache's share is the only one that has grown over the past three years. This is a big deal. The fact that BSD may not run on your PC is not important: The desktop is like the steering wheel to your car, not the engine. The engine is the Internet, and increasingly it is built around Open Source software.

Beyond its importance for the software industry and Internet economy, Open Source software is an interesting social phenomenon with broader economic implications. Two questions are particular puzzling for social scientists. The first concerns the micro-motivations of individuals in the Open Source software community – why do programmers contribute when they obtain no direct remuneration? Open Source software presents a collective action problem, since there are no clear individual incentives for participation. Part of the answer is cultural, sharing is a cultural norm in the software community, and software engineers see themselves as artists who are proud to show others the creative genius of their programming work.

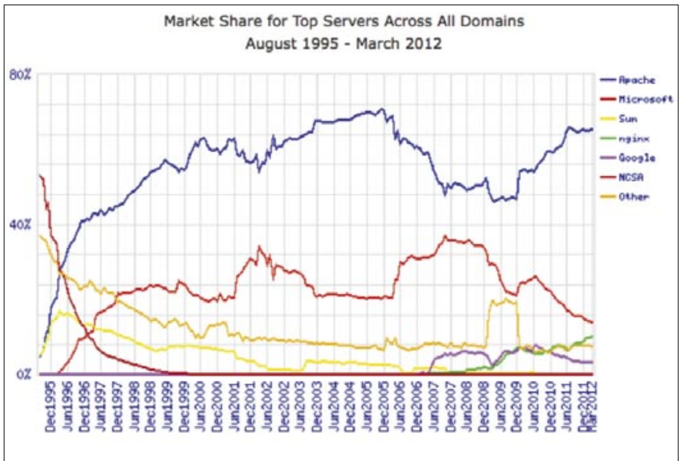
There are real economic incentives for participating in Open Source software development. A reputation as a good programmer has great value for employees of commercial software firms, who may seek to change jobs in the future – yet the product of their paid work is only visible to a few other individuals at their current employer, and they receive no public credit for their efforts. In contrast, the credits file of a piece of Open Source software lists who contributed what to the final product, and potential employers can examine the source code to judge the contribution for themselves. I firmly believe writing Open Source code is probably the most efficient way to establish a reputation as a great programmer. And a reputation as a great programmer is something that you can make money on in other settings. In fact, this reputation-enhancing incentive lends an element of self-selection to the Open Source process, resulting in a better end product. The best programmers actively seek to showcase their work, while mediocre programmers would not want to bear their sub-par code to public scrutiny.

A second interesting question about the development of Open Source software concerns the macro-foundations of the movement – how does the group manage to coordinate the efforts of as many as 7,000 contributors? First, the development of Open Source software enjoys positive network externalities – it benefits from the participation of greater and greater numbers of individuals, even they do not directly contribute code to include in future versions. Every Open Source "free rider" now becomes at minimum a beta tester, who can find bugs, identify a new feature, and otherwise contribute to the collective good. The more copied and widely used a piece of Open Source software, the more valuable it becomes.

Second, Open Source programmers divide a piece of software into a number of small, self-contained modules, each of which can be developed and perfected without knowing precisely how other modules function. Many contributors can thus work on different aspects of the project in parallel, minimizing the number of colleagues with whom each programmer must coordinate. Finally, the Open Source movement may have the outside appearance of an anarchic community, but in reality it includes formal decision-making structures and a set of shared norms to coordinate efforts and govern interaction.

While the Open Source method is now actively revolutionizing the software industry, the mechanics of Open Source production have potentially much broader implications for the economy as a whole. The key to seeing beyond the software industry is to think of Open Source as a production process, where the software is simply a side-effect. There are four observations as to what this process tells us about the broader economy:

First, Open Source is yet another demonstration of how the Internet facilitates geographically widespread



| Developer | February 2012 | Percent | March 2012 | Percent | Change |
|---|---|---|---|---|---|
| Apache | 397,867,089 | 64.92% | 420,337,139 | 65.24% | 0.32 |
| Microsoft | 88,210,995 | 14.39% | 88,971,973 | 13.81% | -0.58 |
| nginx | 60,627,200 | 9.89% | 65,369,149 | 10.15% | 0.25 |
| Google | 19,394,196 | 3.16% | 21,150,938 | 3.28% | 0.12 |

**Figure 1.** *Retrieved from Netcraft on March 17, 2012: http://news.netcraft.com/archives/2012/03/05/march-2012-web-server-survey.html#more-5719*

collaboration, reducing the costs of communication across great distances.

Second, the movement has shown the value of distributed innovation and offers a new way to think about the division of labor in other industries.

Third, Open Source software offers new ideas as to how to combine communities and commerce. The open software community is a relatively unique example of an open, value-driven community where participation is directed towards the creation of a significant product that travels outside the community itself. The model which has succeeded here may have potential in other areas of the economy.

Finally, Open Source software is going to be a significant factor in the international economy in the 21st century. The production of BSD operating systems has involved programmers from around the globe, many of them from developing countries that have security or economic reasons to avoid using proprietary software. I think you can spin out an interesting story where Open Source software turns out to be a powerful instrument of development bootstrapping. Open Source software shifts the decision making prerogative into the hands of people in developing countries.

In this past, Microsoft's reaction to the Open Source software movement bears resemblance to the reaction of national telecommunications carriers when the Internet began to emerge and threaten their entrenched positions. As in this earlier struggle, stakeholders have a lot of influence with policy makers, and they have the potential to leverage this influence into decisions that would be detrimental to the Open Source software movement. Microsoft poses a formidable challenge to the Open Source software movement.

The company's first move was to deny that Open Source posed any real threat to its interests, but it has since shifted to a two-pronged strategy: spreading fear, uncertainty, and deception about Open Source (the FUD factor); and embracing and extending the concept to render it less threatening. This latter tactic is epitomized by Microsoft's shared source initiative, in which it proposes to share its code with other large software companies but not allow them to modify it or distribute it further. Through use of these tactics, it would be easy for Microsoft to sow a lot of confusion about Open Source and get the policy community on its side. If it succeeded in changing copyright law or using patents to prevent reverse engineering of software products, for instance, this could have dire consequences for the Open Source movement.

Let's address security of Open Source versus proprietary software. I have confidence that Open Source software would be more secure than proprietary software because of the greater number of programmers working to expose and correct security flaws. A security problem in software is like a bug; it will be more transparent in Open Source software and therefore fixed more quickly. Some people, such as Alexis de Tocqueville Institution, have suggested that Open Source programmers (or terrorists) could intentionally build back doors (areas of security weakness that the programmers could exploit) into the modules they were designing. However, now that Open Source software is beginning to exist in a corporate setting there is greater incentive to control the security of the whole product.

Now let's look at Open Source software as a business model. Open Source software is currently popular only among more technically-adept computer users. How well would it fare among those that are less tech-savvy? This problem is slowly being solved in the marketplace by people at KDE and GNOME, who build user-friendly interfaces and provide customer service for Open Source software. Furthermore, all end users will appreciate the value of software that has fewer bugs and is less likely to crash. Another legitimate question is how members of the Open Source community (who write and perfect the software without compensation) feel about other companies making money off of their product. Most everything sold by companies is itself Open Source software. They are thus abiding by the ethic of the community, which stresses access to the source code but does not prohibit making money.

Finally, there is the potential of Open Source software production in the developing world. At some point, you have to go beyond operating systems and develop applications; for this task people may have to be paid. In the developing world, software export is driven by an army of paid programmers. How well will Open Source work in this environment? Companies in the developing world can build proprietary applications for Open Source operating systems; this would be an effective model in developing countries that prefer Open Source operating systems for economic or security reasons.

Many developing countries now train students with Open Source software because it is affordable and it allows students to understand how software works on the inside. Once they understand an Open Source system, it may be argued that these students can easily build proprietary applications to run on top of it.

**PAUL AMMANN**

***Paul Ammann lives in New Fairfield, CT with his wife and 4 cats. You can reach him at http://bsdday.eu/2012.***

# Great Specials

## On FreeBSD & PC-BSD Merchandise

Give us a call & ask about our
**SOFTWARE BUNDLES**
**1.925.240.6652**

**$39.95**
FreeBSD 9.0 Jewel Case CD Set
or FreeBSD 9.0 DVD

**$29.95**
PC-BSD 9.0 DVD

**$49.95**
Save a BUNDLE!
The PC-BSD 9.0 Users Handbook
PC-BSD 9.0 DVD

**$99.95**
The FreeBSD CD **or** DVD Bundle
**Inside each CD/DVD Bundle, you'll find:**
FreeBSD Handbook, 3rd Edition
Users Guide FreeBSD Handbook, 3rd Edition, Admin Guide
FreeBSD 9.0 CD or DVD set
FreeBSD Toolkit DVD

*Stylish Dress Attire*
Look Your Professional Best

*Comfy Hoodies*
Stay Warm in Pullovers & Zip Ups

*T-Shirts*
Lots of Styles to Choose From

### FreeBSD 9.0 Jewel Case CD/DVD.............................. $39.95

CD Set Contains:

- **Disc 1**: Installation Boot LiveCD (i386)
- **Disc 2**: Essential Packages Xorg, GNOME2 (i386)
- **Disc 3**: Installation Boot LiveCD (amd64)
- **Disc 4**: Essential Packages Xorg, GNOME2 (amd64)

FreeBSD 8.2 CD.............................................................$39.95
FreeBSD 8.2 DVD...........................................................$39.95

### FreeBSD Subscriptions

Save time and $$$ by subscribing to regular updates of FreeBSD

FreeBSD Subscription, start with CD 9.0..........................$29.95
FreeBSD Subscription, start with DVD 9.0........................$29.95
FreeBSD Subscription, start with CD 8.2..........................$29.95
FreeBSD Subscription, start with DVD 8.2........................$29.95

### PC-BSD 9.0 DVD (Isotope Edition)

PC-BSD 9.0 DVD ...........................................................$29.95
PC-BSD Subscription......................................................$19.95

### The FreeBSD Handbook

The FreeBSD Handbook, Volume 1 (User Guide) ....................$39.95
The FreeBSD Handbook, Volume 2 (Admin Guide)................$39.95

### The FreeBSD Handbook Specials

The FreeBSD Handbook, Volume 2 (Both Volumes)...............$59.95
The FreeBSD Handbook, Both Volumes & FreeBSD 9.0 ........$79.95

### PC-BSD 9.0 Users Handbook ........................................ $24.95

### BSD Magazine.................................................................. $11.99

### The FreeBSD Toolkit DVD ............................................... $39.95

### FreeBSD Mousepad ........................................................ $10.00

### FreeBSD & PCBSD Caps .................................................. $20.00

### BSD Daemon Horns ..........................................................$2.00

*Bundle Specials!*
Save $$$

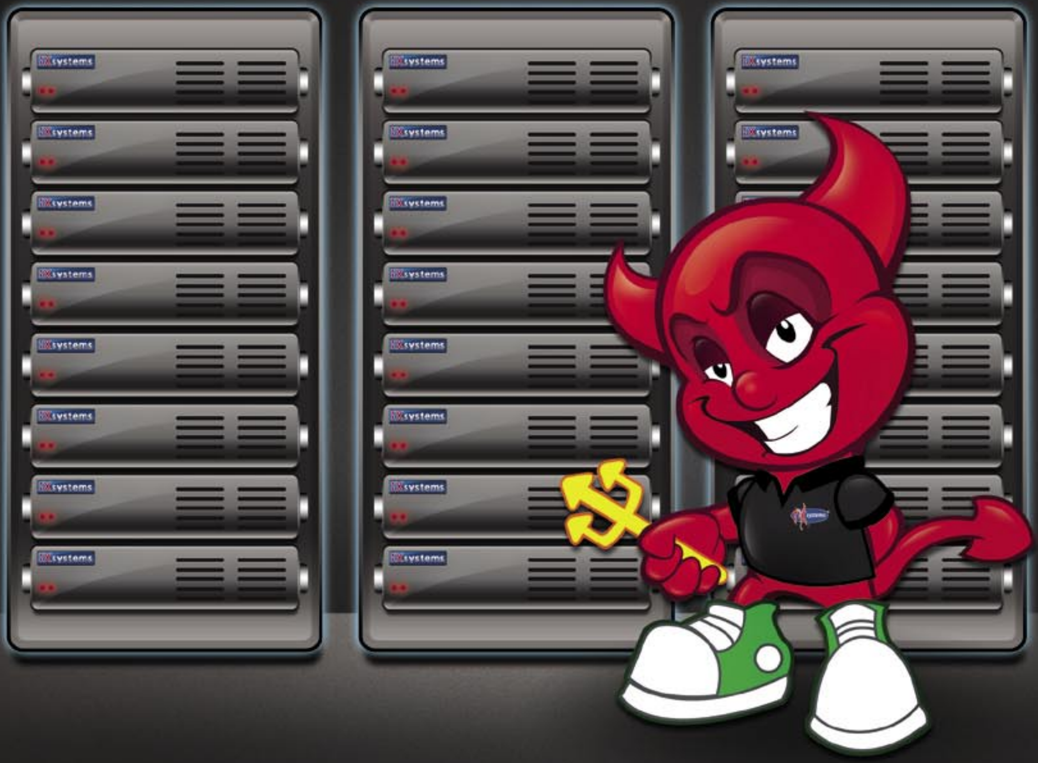
*Just Plain Fun*
Mousepads & Novelty Horns

*BSD Magazine*
Available Monthly

FreeBSD Mall

For even **MORE** items
visit our website today!
**www.FreeBSDMall.com**

## What has your server vendor done for BSD lately? Probably, not much.



## Work with a vendor that supports the operating system you love!

iX is the corporate sponsor of the PC-BSD® Project, a major corporate donor to the FreeBSD Foundation, and leads the FreeNAS™ development team -- all while employing some of the most brilliant minds in the FreeBSD® community. For BSD hardware and software expertise, look no further.

1-855-GREP-4-IX
http://www.iXsystems.com/community

iX systems®